COMPUTACION
Unidades 0 y 1

Cátedra de Computación

BIBLIOTECA DEL
CICLO BASICO

1





EUDEBA S.E.M.
Fundada por la Universidad de Buenos Aires

EDITORIAL UNIVERSITARIA DE BUENOS AIRES
Sociedad de Economía Mixta
Rivadavia 1571/73
Hecho el depósito que marca la ley 11.723
ISBN 950-23-0142-0
IMPRESO EN LA ARGENTINA

UNIDAD 0

A- GENERALIDADES

1 - GENESIS

El lenguaje "TIMBA" (Terribly Imbecile Machine for Boring Algorithms) fue desarrollado por un equipo de trabajo de la Universidad Nacional de San Luis, dirigido por el profesor Ing. Hugo Ryckeboer, como respuesta a la necesidad de contar con un lenguaje para la enseñanza de programación.

2 - EL MAZO

Antes de proceder a describir TIMBA, deberá quedar muy en claro qué es ùn mazo de cartas, porque para trabajar con TIMBA, por supuesto, hay que entender de cartas...

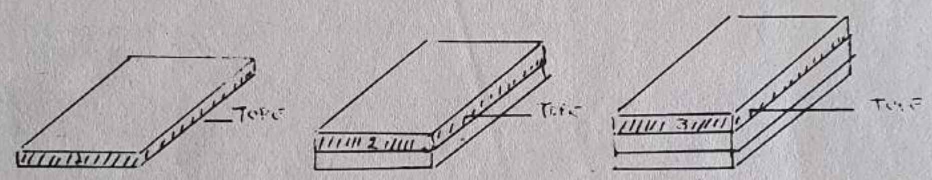
Un mazo de cartas (o pila de cartas, serán sinónimos en este manual) es un conjunto de cartas colocadas una encima de la otra (apiladas).

Todos tenemos noción de lo que es una pila: pila de platos, de ladrillos, de cartas, de libros... En todos los casos unos están encima de los otros, como formando un torre.

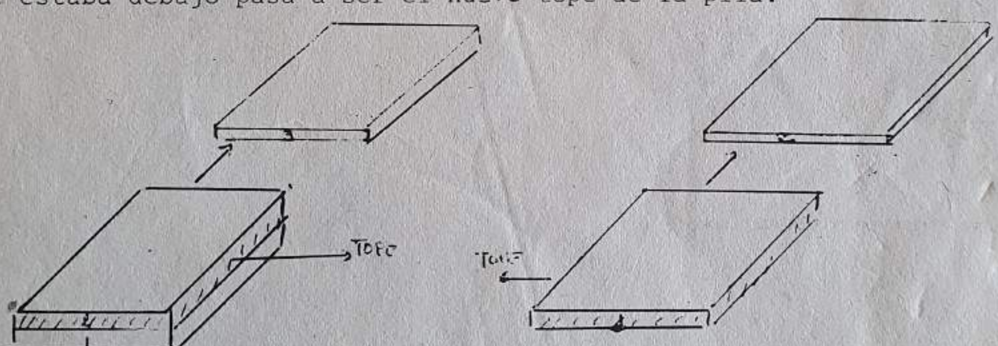
Pero, volvamos a lo que nos ocupa, las pilas de naipes.

Una pila, salvo que esté vacía, tendrá siempre una carta por encima de las demás. Esta carta, aunque sea una sola, se llamará tope de pila.

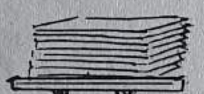
Cada carta que se incorpore a la pila se debe colocar sobre el tope, y al hacerlo se convierte en el nuevo tope de la pila.



Al retirar cartas de una pila estamos restringidos a hacerlo de una por vez. La carta que retiramos es, siempre, el tope y entonces la carta que estaba debajo pasa a ser el nuevo tope de la pila.



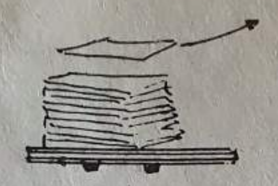
Se podrá tener más de un mazo de cartas, tantos como se deseen



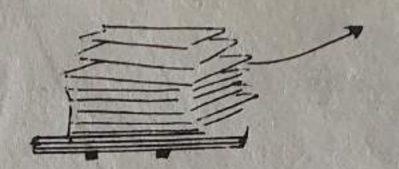
Esto es una pila de cartas.



Esto también es una pila. Como no tiene cartas, diremos que está vacía.



Esto se puede hacer.



Pero esto no.

3 - EL LENGUAJE "TIMBA"

"TIMBA" es un lenguaje de programación que maneja pilas de cartas:permite describir cómo está compueso el mazo, tomar y agregar cartas en un mazo e invertir la carta que en ese momento se analiza.

Desde el punto de vista del programador, un programa en "TIMBA" es una secuencia de órdenes a un ejecutor llamado UCP, que entiende TIMBA, y obedece esas órdenes (si puede) o protesta si no está en condiciones de obedecer (si hay algo que no entiende o que no puede hacer). UCP reconoce las pilas por su nombre, nombre que les fue dado en el propio programa, como veremos, las órdenes a UCP son de la forma:

"TOMAR" : toma la carta que está en el tope de una pila y la retiene en la mano.

"DEPOSITAR" : deposita la carta de su mano en el tope de una pila.

"INVERTIR" : invierte la carta que tiene en la mano de tal modo que si estaba boca abajo, pasa a estar boca arriba y viceversa. (UCP continúa reteniendo la carta)

B - PROGRAMA

Un programa "TIMBA", consta de dos partes: una primera parte que contiene una secuencia de órdenes a UCP, que describen el proceso que se quiere realizar, y, una segunda parte con la descripción (o datos) de las pilas.

1 - SENTENCIAS

Llamaremos sentencias a las órdenes a UCP. Las hay de tres tipos:

1.1 Operativas.
1.2 De selección.
1.3 Iterativas.

1.1 Las sentencias operativas se refieren a acciones que se toman sobre las pilas o cartas. Las órdenes posibles son:

a) TOME UNA CARTA DE LA PILA nombre de la pila.

b) DEPOSITE CARTA EN LA PILA nombre de la pila.

c) INVIERTA LA CARTA

El orden en el que se ejecutan las sentencias, es el de la secuencialidad natural (o sea que si dos sentencias se escriben una a continuación de la otra, se ejecutarán una a continuación de la otra)

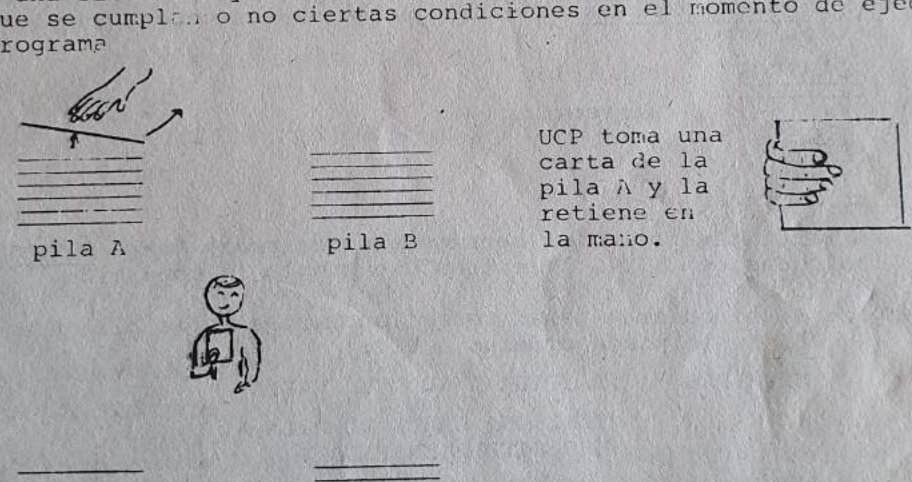
Si, por ejemplo, se escribe:

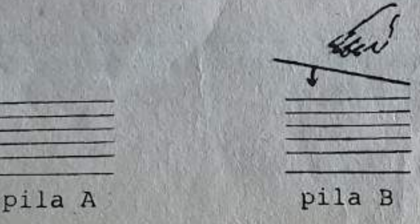
(1) TOME CARTA DE LA PILA A ,

(2) DEPOSITE CARTA EN LA PILA B

UCP ejecutará primero la orden de tomar una carta de A y luego la depositará en B.

1.2 Existen además sentencias de selección que permiten ordenar a UCP que evalúe una situación y en función de ella, adopte alternativas de acuerdo a que se cumplan o no ciertas condiciones en el momento de ejecución del programa





UCP deposita carta en pila B.

La estructura de las secuencias de selección es la siguiente:

(0) SI condición
(1) sentencias

SINO
(2) sentencias

NADA MAS

(3) sentencias

O sea, que si en tiempo de ejecución, la condición se cumple, UCP ejecuta las sentencias de (1) y si no se cumple ejecuta las sentencias de (2). Cuando termina de ejecutar o bien (1), o bien (2), sigue siempre con (3).

Por ejemplo:

Sentencia de Selección:

```
SI         PILA A ESTA VACIA           (0)
           TOME CARTA DE PILA B        (1)
SINO
           TOME CARTA DE PILA A        (2)
NADA MAS
DEPOSITE CARTA EN PILA C               (3)
```

Con estas sentencias se logra lo siguiente:
Si no hay cartas en A, tomarla de B; pero si hay cartas en A, tomarla de A. Una vez que UCP tiene una carta en la mano, venga de A o de B, la deposita en la pila C.

1.3 Las Sentencias Iterativas permiten ordenar a UCP la realización de un bloque de sentencias una, ninguna o varias veces. Su estructura es la siguiente:

(0) MIENTRAS condición
(1) sentencias

REPITA

(2) sentencias

Aquí se repetirán todas las sentencias encerradas entre MIENTRAS y REPITA (indicadas con (1)), "mientras" se cumpla la condición.

Por ejemplo: Si se quieren pasar todas las cartas de la pila A a la pila B, se le puede ordenar a UCP:

```
MIENTRAS     LA PILA A NO ESTA VACIA      (0)
             TOME UNA CARTA DE PILA A,  }
             DEPOSITELA EN PILA B       } (1)
REPITA
```

Si la pila A está vacía de entrada, el bloque (1) no se ejecuta. En cambio si hay cartas en A, el bloque (1) se ejecutará hasta que A se vacíe.

Para escribir un programa TIMBA, se procederá de la siguiente manera: Colocamos el encabezamiento (Que indicará a UCP que comienza un programa)

DEFINICION DE PROGRAMA luego se indicarán todas las sentencias que lo lo describen (operativas, de selección, iterativas), luego un punto y coma (;), y a conti-

nuación las palabras:

UCP EJECUTE CON LAS SIGUIENTES CARTAS: Se indica el comienzo de los datos.

Luego se describen las pilas de cartas con las que ejecutará UCP.

2. DATOS : Se describe una pila por vez, separadas unas de las otras por una coma. Después de la última, va un punto(.).
Una pila puede estar vacía (sin cartas) o bien tener cartas.
Cuando una pila está vacía se la describe siempre de la siguiente forma:

LA PILA nombre de pila NO TIENE CARTAS.

Por ejemplo: LA PILA A NO TIENE CARTAS.
Cuando una pila tiene cartas, se debe describir su contenido. Una carta se describe por su palo, su número y si está boca arriba (↑) o boca abajo (sin flecha).

Por ejemplo: PILA B TIENE 2 DE OROS - 3 DE BASTOS ↑ - 5 DE COPAS.

Con todos estos elementos, escribamos ahora un programa TIMBA que dada una pila de cartas separe las cartas de la pila en dos pilas: una que contenga todas las de la pila original que hayan estado boca arriba y otra con las que hayan estado boca abajo.

Llamemos a la pila original NAIPES, a la que tendrá todas las cartas boca abajo TAPADAS, y a la que tendrá todas las cartas boca arriba DESCUBIERTAS.

Se tratará de ir tomando una por una las cartas de la pila NAIPES hasta que se agoten; y si la carta está boca abajo, depositarla en la pila TAPADA, sino depositarla en DESCUBIERTAS.

```
DEFINICION DE PROGRAMA
    MIENTRAS LA PILA NAIPES NO ESTA VACIA
            TOME UNA CARTA DE LA PILA NAIPES
            SI LA CARTA ESTA BOCA ABAJO
               DEPOSITELA EN LA PILA TAPADAS
            SINO
               DEPOSITELA EN LA PILA DESCUBIERTAS
            NADA MAS
    REPITA;
UCP EJECUTE CON LAS SIGUIENTES CARTAS:
    LA PILA NAIPES TIENE 2 DE OROS - 3 DE BASTOS ↑ - 5 DE COPAS -
    7 DE ESPADAS - 12 DE BASTOS ↑ ,
    LA PILA TAPADAS NO TIENE CARTAS ,
    LA PILA DESCUBIERTAS NO TIENE CARTAS.
```

Por supuesto podemos poner en la pila NAIPES cualquier combinación de cartas y el programa siempre separará como pedimos. Las pilas TAPADAS y DESCUBIERTAS siempre estarán vacías al empezar.

TOMEN UN MAZO DE CARTAS Y SIGAN LAS INSTRUCCIONES....

C - DESCRIPCION FORMAL DEL LENGUAJE TIMBA

1 - NOTACION

Como hemos visto, para manipular los mazos debemos escribir un programa que contiene órdenes para UCP. Pero UCP es demasiado limitado y no puede entender mensajes ambiguos. Por ejemplo, si tenemos dos pilas A y B y UCP tiene una carta en la mano, él no podra realizar una orden tan simple como DEPOSITE LA CARTA EN LA PILA QUE ESTA A SU IZQUIERDA, debido a ello, debemos precisar la manera en que se deben expresar las sentencias en TIMBA.

Existen una serie de maneras para formalizar la escritura de sentencias (sintaxis) en lenguajes de programación.

Aqui usaremos una de ellas

En lo que sigue aparecerán tres elementos: corchetes ([ ]), llaves ({ }) y asterisco (*) con los siguientes significados.

1.1 Corchetes: el programador podrá optar por escribir o no aquello que está encerrado entre corchetes, sin que cambie el sentido de la frase. Por ejemplo:

TOME [UNA] [CARTA] DE [LA] PILA nombre de pila

significa que son igualmente correctas, y tienen el mismo significado las siguientes frases:

(1) TOME DE PILA A

(2) TOME DE LA PILA A

(3) TOME CARTA DE PILA A

(4) TOME CARTA DE LA PILA A

(5) TOME UNA CARTA DE PILA A

(6) TOME UNA CARTA DE LA PILA A

(7) TOME UNA DE PILA A

(8) TOME UNA DE LA PILA A

1.2 Llaves: Cuando en una descripción aparecen dos o más items entre llaves, el programador debe elegir entre ellos, aquel que se ajuste a su problema, o, si son equivalentes, aquel que prefiera.

Por ejemplo:

$$\left\{ \begin{array}{l} \text{INVIERTA LA CARTA} \\ \text{INVIERTALA} \end{array} \right\}$$

Para invertir una carta deberá elegir entre la sentencia INVIERTA LA CARTA y la sentencia INVIERTALA.

1.3 Asterisco : Un par de corchetes seguido de un asterisco significa que el item encerrado entre corchetes se puede repetir. Por ejemplo, vimos que la sentencia iterativa MIENTRAS era:

```
MIENTRAS   condición
           sentencias
REPITA
```

Podemos escribirlo:

```
MIENTRAS   condición
           sentencia [sentencia]*
REPITA
```

Esto significa que escribiremos al menos una sentencia, pero que podemos escribir todas las que necesitemos según nuestro problema

Ahora describiremos todos los elementos del lenguaje TIMBA formal mente.

I - SENTENCIAS OPERATIVAS

Las sentencias operativas son tres, formalmente:

TOME [UNA [CARTA]] DE [LA] PILA nombre de pila

ó

{ DEPOSITE LA CARTA / DEPOSITELA } EN [LA] PILA nombre de pila

ó

{ INVIERTA LA CARTA / INVIERTALA }

la ejecución por UCP de una sentencia operativa TOME presupone que la pila cuyo nombre figura en la sentencia ha sido definida en la segunda parte del programa. La carta que figura al tope de la pila es tomada por UCP, y cualquier referencia posterior que se haga al ente CARTA se interpretará como una referencia a la última carta tomada por UCP (siempre que no hubiere sido depositada).

DEPOSITE ordena a UCP dejar la carta que tiene en la mano en ese momento, en una pila determinada.

La pila debe ser declarada en la segunda parte del programa. La ejecución de un DEPOSITE presupone que UCP tiene, efectivamente, una carta en la mano; lo que se consigue por medio de la ejecución de una sentencia TOME.

Por obvias razones, UCP detectará un error de ejcución si no tuviera una carta en la mano y se le ordenara DEPOSITE. Por lo mismo, UCP no admite la ejecución de dos TOME sin un DEPOSITE entre ellos, detectando también error de ejecución.

Por último, UCP reconoce error de ejecución al intentar tomar una carta de una pila vacía.

El verbo INVIERTA se refiere a la carta que UCP tiene en la mano. Esta reconoce dos estados: NO BOCA ABAJO y BOCA ABAJO. La ejecución por UCP de una sentencia INVIERTA altera el estado de CARTA: si éste era NO BOCA ABAJO pasa a ser BOCA ABAJO y viceversa.

El verbo INVIERTA presupone la existencia de carta, sino UCP detecta un error de ejcución.

Mensajes de error:

Ej. 1: ME PIDE UD. QUE TOME DE PILA nombre, QUE ESTA VACIA.

Ej. 2: ME PIDE UD. QUE TOME DE PILA nombre, Y YO YA TENGO UNA CARTA: el número de palo.

Ej. 3: ME PIDE UD. QUE DEPOSITE EN PILA nombre, Y YO NO TENGO CARTA.

Ej. 4: ME PIDE UD. QUE INVIERTA LA CARTA? PERO YO NO TENGO CARTA.

II - SENTENCIA DE SELECCION SI

La sentencia SI permite al programador solicitar a UCP que evalúe una condición y proceder, de acuerdo con ella, a la ejecución de un bloque u otro, por lo que la secuencia normal de ejecución puede depender de las condiciones halladas en las pilas, esto es, de la situación o el estado de las pilas.

Formalmente una sentencia SI debe escribirse:

SI condición sentencias { SINO sentencias NADA MAS / SINO NADA MAS }

Si la condición (ver CONDICION) es verdadera en el momento de analizarla UCP, éste procederá con la ejecución del primer bloque de sentencias, esto es, las sentencias que preceden al SINO.
Si la condición fuese falsa, UCP procederá con la ejecución de las sentencias que hubiere entre el SINO y el NADA MAS, y si no las hubiera, con la sentencia que sigue a la sentencia SI? es decir, la que empieza despues de la coma (,) que sigue a NADA MAS.
Las sentencias que puede escribirse en un SI son todas las de "TIMBA", por lo tanto, un SI puede contener otro SI (y aún un MIENTRAS).
Tal estructura de un SI dentro de otro se llama usualmente "Nidos de SI". (ver SENTENCIAS).
Ej. :

SI LA PILA A ESTA VACIA TOME UNA DE PILA B SINO TOME DE PILA A NADA MAS, DEPOSITELA EN PILA C.

UCP verificará si hay cartas en la pila A, y procederá, de acuerdo con ello, a tomar una carta de A o de B, si A fuera vacía. En ambos casos la carta es depositada en el tope de C.

III - SENTENCIA DE ITERACION MIENTRAS

La sentencia MIENTRAS permite al programados ordenar a UCP la realización repetida y condicionada de un bloque de sentencias, gobernado por una condición verificable en la ejecución. (ver CONDICION).
Formalmente, la sentencia MIENTRAS debe escribirse:

MIENTRAS condición sentencias REPITA

Cuando UCP encuentra durante la ejecución de un programa una sentencia MIENTRAS, evalúa primero la condición (ver CONDICION) y si ésta es verdadera, ejecuta normalmente las sentencias que siguen hasta el REPITA. Al hallar el REPITA, vuelve a evaluar la condición, volviendo a ejecutar todas todas las sentencias si la condición es aún verdadera, y así siguiendo hasta que la condición resulte falsa.
Cuando la condición resulta falsa, ya sea en la primera, segunda, ..., n-ésima iteración, UCP se salta la ejecución de las sentencias hasta el REPITA, prosiguiendo con la sentencia que sigue al MIENTRAS, esto es lo que comienza después de la coma (,) que sigue al REPITA.

Ej.

MIENTRAS LA PILA B NO ESTA VACIA TOME DE PILA B, DEPOSITE EN PILA C REPITA, TOME DE PILA D

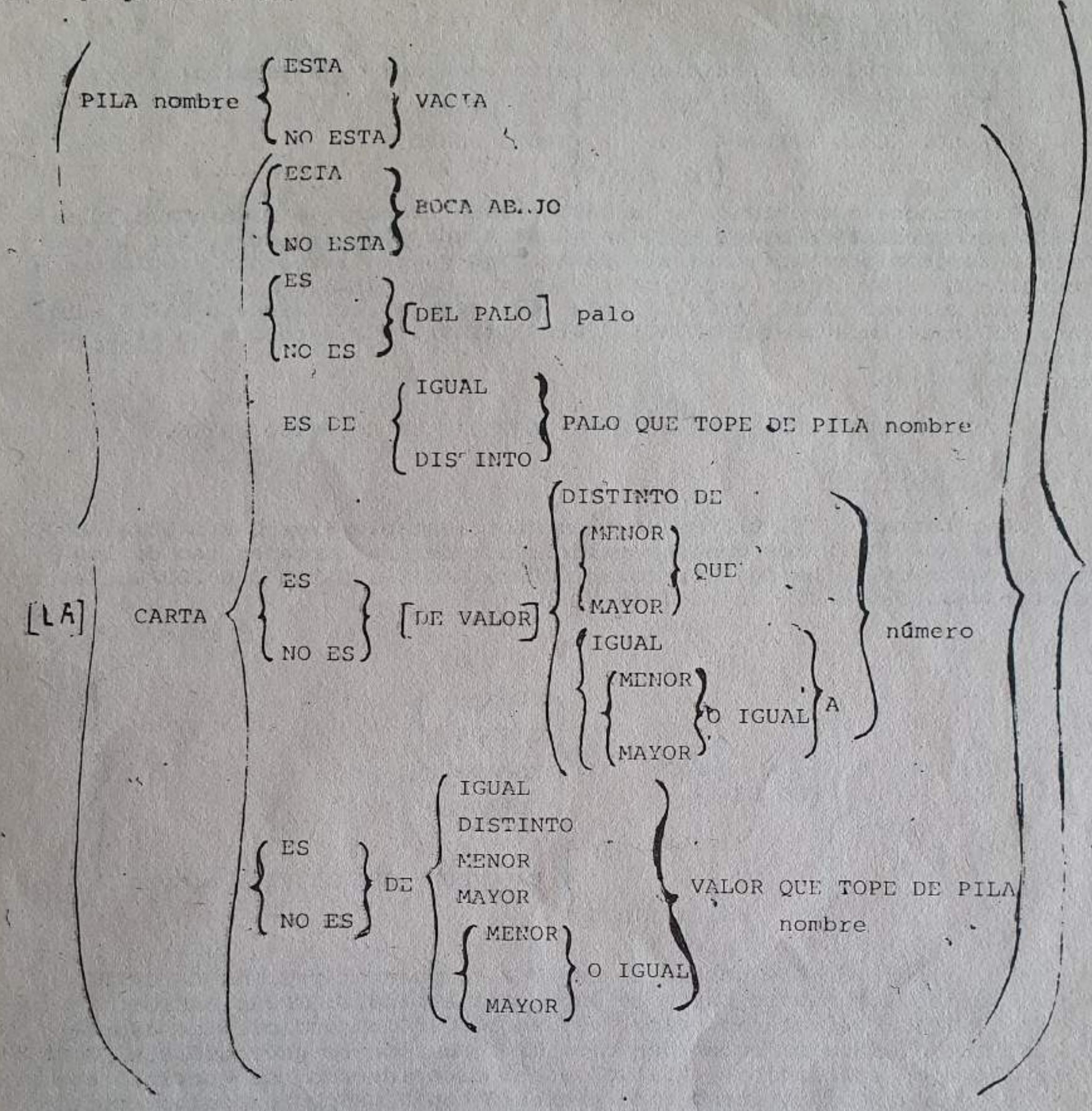
UCP Vaciará la pila B, pasando a la pila C las cartas de B de a una. Una vez vacía la pila B, UCP tomará la carta del tope de la pila D.

IV - CONDICION

Una condición en "TIMBA" es una proposición lógica, simple o compuesta, que emite un juicio sobre el estado de las pilas o la carta, y que puede por lo tanto ser evaluado como "verdadero" o "falso".
Formalmente, una condición en "TIMBA" es:

$$\text{proposicion} \left[\begin{Bmatrix} Y \\ O \end{Bmatrix} \text{proposicion} \right]^*$$

Donde proposición es:



V - PILA VACIA

Formalmente, una condición de PILA VACIA se escribe:

[LA] PILA nombre { ESTA / NO ESTA } VACIA

UCP reconoce si una pila está vacía cuando no tiene cartas en la pila. Si hubiera una, dos o más cartas en la pila, PILA nombre ESTA VACIA sería reconocido como "falso" por UCP, mientras que PILA nombre NO ESTA VACIA sería "verdadero". UCP reconoce un error de ejecución si se intenta tomar cartas de una pila vacía. Por lo tanto, la condición de pila vacía permite construir sentencias que eviten tomar de pilas sin cartas.

VI - CARTA BOCA ABAJO

Formalmente, una condición de carta boca abajo se escribe:

[LA] CARTA { ESTA / NO ESTA } BOCA ABAJO

UCP reconoce dos estados a la CARTA: BOCA ABAJO y NO BOCA ABAJO. Si la CARTA estuviera BOCA ABAJO, UCP no puede hacer comparaciones, por lo que esta condición permite construir sentencias que la inviertan condicionalmente.

Si no hubiera CARTA definida para UCP, esto es, si ninguna CARTA hubiere sido depositada (ver SENTENCIAS OPERATIVAS), UCP reconoce un error de ejecución.
Mensaje de error:

Ej. 5 UD. QUIERE SABER COMO ESTA LA CARTA, Y YO NO TENGO CARTA.

VII - PALO DE CARTA

Una carta en "TIMBA" es un elemento biestable (ver CARTA BOCA ABAJO) con dos atributos :palo y valor. Palo de una carta es uno de los cuatro palos de la baraja española : OROS, BASTOS, ESPADAS o COPAS. Valor es un número del 1 al 7 o´ del 10 al 12

[LA] CARTA {
{ ES / NO ES } [DEL PALO] { OROS / BASTOS / ESPADAS / COPAS }
ES DE { IGUAL / DISTINTO } PALO QUE TOPE DE PILA nombre
}

En la primera de las dos opciones, UCP compara el palo de LA CARTA contra uno de los cuatro palos de la baraja española: OROS, BASTOS, ESPADAS, COPAS; detectando valor "verdadero" si la condición se escribió con ES y los palos coinciden o si la condición se escribió con NO ES y los palos no coinciden; y "falso" en el caso inverso: se escribió con ES y los palos son distintos o se escribió con NO ES y los palos coinciden. UCP detecta un error de ejecución si no tiene CARTA (ver SENTENCIAS OPERATIVAS) o si ésta estuviere BOCA ABAJO (ver CARTA BOCA ABAJO).

En la segunda opción, UCP compara el palo de la CARTA contra el palo de la carta que está al tope de la pila "nombre", detectando un valor "verdadero" si la condición se escribió con IGUAL y los palos coinciden o si se escribió con DISTINTO y los palos no coinciden, siendo para UCP "falsa" la condición si se escribió ésta con DISTINTO y el palo es el mismo.

UCP detecta un error de ejecución si no tiene carta en la mano (ver SENTENCIAS OPERATIVAS) o si cualquiera de las dos esta BOCA ABAJO (ver CARTA BOCA ABAJO), o la pila ESTA VACIA (ver PILA VACIA).
Mensajes de error:

Ej. 6 UD. QUIERE SABER SI LA CARTA ES DE palo, Y YO NO TENGO CARTA.

Ej. 7 UD. QUIERE SABER SI LA CARTA ES DEL PALO DEL TOPE nombre, Y YO NO TENGO CARTA.

Ej. 8 UD. QUIERE SABER SI LA CARTA ES DEL PALO DEL TOPE DE nombre, PERO LA CARTA ESTA BOCA ABAJO.

Ej. 9 UD. QUIERE SABER SI LA CARTA ES DEL PALO DEL TOPE DE nombre, PERO EL TOPE ESTA BOCA ABAJO.

Ej. 10 UD. QUIERE SABER SI LA CARTA ES DEL PALO DEL TOPE DE nombre, PERO LA PILA ESTA VACIA.

Ej. 17 UD. QUIERE SABER SI LA CARTA ES DE palo PERO LA CARTA ESTA BOCA ABAJO.

VIII - <u>VALOR DE CARTA</u>

Una carta en "TIMBA" es un elemento biestable (ver CARTA BOCA ABAJO) con dos atributos :palo y valor. Palo de una carta es uno de los cuatro palos de la baraja española: OROS, BASTOS, ESPADAS o COPAS. Valor es un número del 1 al 7 ó del 10 al 12.
Formalmente, la condición de valor de carta se escribe:

```
                                         DISTINTO DE
                                        (MENOR
                                        (        QUE
                   ES                   (MAYOR             número
                  (        [DE VALOR]   IGUAL
                   NO ES                  MENOR
                                         (       O IGUAL  A
                                          MAYOR
[LA] CARTA
                          IGUAL
                          DISTINTO
                   ES     MENOR
                  (    DE MAYOR               VALOR QUE TOPE DE PILA
                   NO ES    MENOR             nombre
                           (      O IGUAL
                            MAYOR
```

En la primera de las dos opciones, UCP compara el valor (1 al 7 ó 10 al 12) de la CARTA contra el "número". La comparación se realiza de acuerdo con las reglas comunes para los números naturales, y cada uno de los operadores relacionales (DISTINTO DE, IGUAL A, MENOR QUE, MAYOR QUE, MENOR O IGUAL A y MAYOR O IGUAL A) representan exactamente lo que significan en lenguaje diario.

UCP reconoce la proposición como "verdadera" si ha sido escrita con la opción ES y la relación se verifica o si ha sido escrita con NO ES y la relación no se verifica. UCP reconoce un error de ejecución si CARTA no está definida, (ver SENTENCIAS OPERATIVAS) o si está BOCA ABAJO (ver CARTA BOCA ABAJO).

En la segunda opción, UCP compara el valor de CARTA, en el sentido de los números naturales, contra la carta al tope de la PILA nombrada.

UCP reconoce la proposición como "verdadera" si ha sido escrita con la opción ES y la relación se verifica o si ha sido escrita con NO ES y la relación no se verifica, UCP reconoce un error de ejecución si CARTA no está definida (ver SENTENCIAS OPERATIVAS), si la PILA ESTA VACIA (ver PILA VACIA) o si por lo menos una de las dos cartas de la comparación esta BOCA ABAJO.

Mensajes de error:

Ej. 11 UD. QUIERE SABER SI LA CARTA ES DE VALOR número, Y YO NO TENGO CARTA.

Ej. 12 UD. QUIERE COMPARAR EL VALOR DE LA CARTA CON EL DEL TOPE DE nombre, Y YO NO TENGO CARTA.

Ej. 13 UD. QUIERE COMPARAR EL VALOR DE LA CARTA CON EL DEL TOPE DE nombre, Y LA PILA ESTA VACIA.

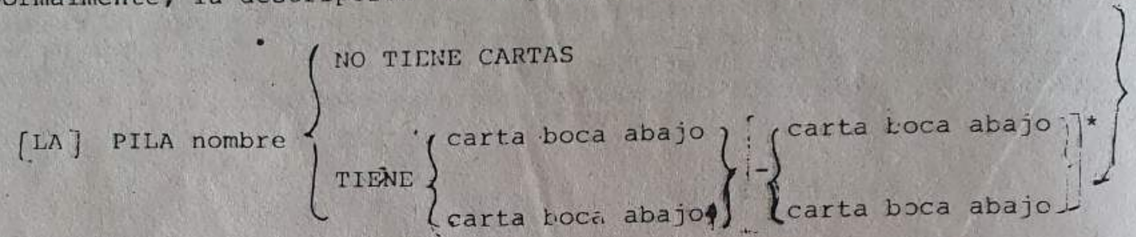
Ej. 14 UD. QUIERE COMPARAR EL VALOR DE LA CARTA CON EL DEL TOPE DE nombre, PERO LA CARTA ESTA BOCA ABAJO.

Ej. 15 UD. QUIERE COMPARAR EL VALOR DE LA CARTA CON EL DEL TOPE DE nombre, PERO EL TOPE ESTA BOCA ABAJO

Ej. 16 UD. QUIERE SABER SI LA CARTA ES DE VALOR número, PERO LA CARTA ESTA BOCA ABAJO.

IX - DESCRIPCION DE PILA

Formalmente, la descripcion de la pila se escribe:



La primera opción hace que UCP asocie al nombre una PILA vacía. La segunda opción hace que UCP asocie al nombre una PILA con las cartas (una, dos ó más) que se describen después de la palabra clave TIENE. La última carta descripta se ubicará al tope de la PILA, la penúltima debajo de ésta y así hasta ubicar todas las cartas.

Todas las cartas son descriptas por su palo y número, y supuestas BOCA ABAJO. Si se desea invertir la carta al definir la PILA, esto es, se desea que la o las cartas aparezcan NO BOCA ABAJO en la PILA, después de cada descripción de carta que se desee NO BOCA ABAJO debe incluirse el símbolo especial;.

Formalmente, carta boca abajo debe escribirse:

```
 1
 2
 3
 4             OROS
 5    DE       BASTOS
 6             ESPADAS
 7             COPAS
10
11
12
```

UCP detecta error de previa ejecución si una PILA utilizada en la descripción del proceso no fuera descripta, y da un mensaje de atención si fuera descripta una PILA no utilizada en la descripción del proceso.

PE 0 LA PILA nombre NO FUE DESCRIPTA.

AT! 1 LA PILA nombre FUE DESCRIPTA SIN NECESIDAD.

X - PALABRAS RESERVADAS DE TIMBA

La siguiente es una lista de palabras reservadas de TIMBA. Estas palabras no pueden ser usadas como nombres de pila.

ABAJO
BASTOS
BOCA
CARTA
CARTAS
CON
COPAS
DE
DEL
DEFINICION
DEPOSITE
DEPOSITELA
DISTINTO
EJECUTE
EN
ES
ESPADAS
ESTA
IGUAL
INVIERTA
INVIERTALA
LA
LAS
MAS
MAYOR
MENOR
MIENTRAS
NADA
NO
O
OROS
PALO
PILA
PROGRAMA
QUE
REPITA
SI
SIGUIENTES
SINO
TIENE
TOME
TOPE
UCP
UNA
VACIA
VALOR
Y
1
2
3
4
5
6
7
10
11
12

PRACTICA 0

Cuando no se aclara el estado de la carta , se supone boca arriba.

1.- Se tienen dos pilas no vacías de cartas (A y B). Generar otra pila C.

a) Donde se encuentren solamente los topes de A y B.

b) Idem pero invertidos.

2.- Dada una pila con al menos dos cartas, redactar un programa en TIMBA que :

a) Intercambie las cartas que ocupan el primero y segundo lugar en la pila

b) (e.d. el tope pasa a ser segunda y la segunda pasa a ser tope).

b) Idem pero dejando el nuevo tope boca abajo.

3.- Dadas 4 pilas de cartas no vacías (como lo indica la figura 1) ,intercambiar sus topes de modo que queden ubicados en orden creciente a partir de la pila A (ver figura 2) a) Utilizando una pila auxiliar .

b) Sin utilizar pilas auxiliares.



4.- Dada una pila A no vacía de cartas boca arriba y/o boca abajo, invertir el tope de la misma si éste está boca abajo (dejar todas las cartas en A).

5.- Sacar de una pila (con al menos dos cartas) la segunda si ésta es de oros.

6.- Dada una pila con al menos 2 cartas, Si el tope es el rey de oros extraer la segunda, en caso contrario invertirla, dejandola en el mismo lugar.

7.- Dadas 3 pilas no vacías A, B y C si los 3 topes son del mismo valor ,extraer las cartas que ocupan el segundo lugar y generar otra pila "SEG" con las mismas (las pilas no necesariamente tienen al menos 2 cartas).

8.- Dada una pila A de cartas pasarlas a otra pila B tal que el tope de A pase a ser la base de B.

9.- Dadas 3 pilas A , B y C de cartas , generar otra pila D donde la pila B está sobre la de A y la C debajo de la A .

a) No se exige que se mantenga el orden en las pilas,no usar pilas auxiliares.

b) Manteniendo el orden en las pilas ,usando solamente una pila auxiliar.

10.- Dada una pila A de cartas ,generar otra pila donde se encuentren solamente los bastos mayores que 3.

111.- Seleccionar de una pila de cartas no vacías, aquellas que sean copas generando una pila con aquellas (dejar la pila A con las cartas restantes como estaban). Una vez ejecutado el programa ¿ Qué circunstancia le indica que en la pila primitiva no había ninguna carta del palo copas ?

12.- Idem al ejercicio anterior, pero las cartas pueden estar boca arriba y/o boca abajo. (No interesa el estado final de las cartas).

13.- Idem al anterior, pero las cartas tienen que quedar en el estado original.

14.- Dada una pila B con al menos 2 cartas extraer el tope y la base de la misma y generar una pila con ellas (las demas cartas deben quedar en el mismo orden en la pila B).

15.- Dada una pila A , no vacía ,de cartas,generar 4 pilas con sólo ases, reyes, sotas y caballos, las restantes artas deben quedar en la pila A.

16.- Dadas 2 pilas de cartas boca arriba y/o boca abajo, intercalar las dos pilas dejando la carta en estado contrario al inicial. (La que estaba boca abajo tiene que quedar boca arriba y viceversa).

<u>Observación</u> : debe intercalarse el mayor número posible de cartas.

17.- Dada una pila A de cartas de la que se sabe que tienen dos ases , extraer de la misma los ases y todas las cartas intermedias. (Las restantes cartas deben quedar en la pila A).

18.- Seleccionar de una pila A las cartas que anteceden a una espada.

19.- Seleccionar de una pila A las cartas que siendo de espadas anteceden a un as.

20.- Dada una pila A , determinar (dejando una señal) si el número de sus elementos es par ó impar . La señal deberá ser una carta invertida.

21.- Dadas las pilas A y CUENTA ," contar" el número de copas de la pila A sin modificar el contenido inicial de A

22.- Dada una pila A , separar las cartas que coinciden en palo con la que sigue. (Otro ídem , pero con la que antecede).

PRACTICA 1

1.- Indicar con dibujos sucesivos los pasos a dar para hacer las siguientes operaciones con un ábaco :

a) 115 + 124 = b) 197 + 288 = c) 134 - 29 =

d) 114 - 88 = e) 402 - 68 =

2.- Indique como resolvería una multiplicación de dos cifras usando las tablas de Nappier.

a) 34 x 43 = b) 88 x 77 =

Dibujar las tablas necesarias para solucionar el problema.

3.- Cómo resolvería , siguiendo el esquema de Hollerit, los siguientes problemas :

a) Clasificar y totalizar un censo según sexo ,edad (mayor - menor) y lugar de vivienda (capital - interior)

b) Idem si deseo clasificar según profesión (obrero - empleado - profesional - campesino), ingresos (altos - medios - bajos) y educación (superior - media - primaria - nula)

En los próximos ejercicios podrán aparecer sobre la cinta de una máquina de Turing distintos símbolos (en los ejemplos del texto ,los símbolos que aparecían eran *, blanco y palote (|), pero es posible que aparezcan tambien letras y/o números , que podrán ser leídos, borrados o escritos en la cinta, del mismo modo que el palote.

4.- En lugar de palotes , usaremos las letras a , b y c . Construir la matriz de una máquina tal que dada una palabra ,permute : a por b

b por c

c por a

(por ejemplo : * a a b a c * se transforma en * b b c b a *)

Aplicar el programa a la palabra : b a c a a c

5.- En lugar de palotes usaremos las letras a y 1

Construir la matriz de una máquina que reconozca palabras capicúas

(Ejemplo : a 1 a es capicúa

a 1 1 a es capicúa

1 a no es capicúa)

Si la palabra es capicúa la máquina para. Si la palabra no es capicúa la máquina da error. Aplicar el programa a las palabras : 1 1 a a

a a 1 a 1 a a

6.- Construir la matriz de una máquina que tome una palabra escrita con las letras a y b, y escriba en la cinta su imagen especular .

(Ejemplo : b a b a se transforma en a b a b)

Aplicar el programa a las palabras : a a b a a b

b a a b a

UNIDAD I

HISTORIA DE LA COMPUTACION HASTA LA ACTUALIDAD

OBJETIVOS

1) Identificar el proceso histórico de construcción de herramientas de cálculo y procesamiento de datos hasta llegar a las modernas computadoras digitales.
2) Distinguir distintos sistemas de numeración, en particular, familiarizarse con el sistema binario y comprender por qué se usa en las modernas computadoras digitales.
3) Operar con una "computadora lógica" hecha con lapiz y papel e introducirse en el concepto de algoritmo.
4) Apreciar al desarrollo de los medios de cálculo como parte del esfuerzo de la sociedad humana por resolver todo tipo de problemas con los que se enfrenta en su devenir histórico.
5) Reflexionar acerca de los mitos creados en torno a la relación entre la computadora y el hombre.

PROGRAMA ANALITICO

- Historia hasta el siglo XIX

  Sistemas de numeración. Herramientas de cálculo. Abaco. Pascal. Leibnitz. Jaquart. Babbage. Hollerith y el procesamiento de datos.

- Historia contemporánea

  La década del 30. Ajuste entre la ciencia y la técnica. Primeros equipos electromecánicos y electromagnéticos. El pensamiento de Turing. El esquema de Von Neumann. Progreso tecnológico. Evolución del Software. Cuestiones de la relación hombre-máquina.

- Anexos:
  - Sistema binario.
  - Babbage: Pionero de la computación automática.
  - Turing: Una computadora hecha con papel y lápiz.
  - La máquina de Von Neumann.

UNIDAD I

HISTORIA DE LA COMPUTACION

INTRODUCCION GENERAL

Esta unidad está dedicada a la evolución de la computación introduciendo simultáneamente al lector a los principios básicos de la computación moderna en su generación histórica. Para ser mas precisos, esta parte del curso está orientada al desarrollo de la computadora digital.

Si bien en general una computadora es una máquina capaz de aceptar datos, realizar con ellos operaciones siguiendo instrucciones (programas) y proveer los resultados de dichas operaciones, existen diversas formas de representar los datos y manipularlos.

Así como un reloj "tradicional" y otro digital cumplen la misma función pero a partir de distintas formas de reflejar el paso del tiempo, también existen en general dos tipos de computadoras: las analógicas y las digitales (en realidad también hay "zonas grises" dadas por equipos híbridos con partes analógicas y otras digitales). A grandes rasgos, un computador analógico representa magnitudes físicas (velocidad, temperatura)por medio de otras magnitudes que son funciones de las primeras. Así, un velocímetro normal, representa la velocidad del móvil mediante ángulos (el que se forma entre la aguja y la base de velocímetro) o un termómetro refleja una magnitud, la temperatura, mediante otra proporcional a la primera, la longitud de la columna de mercurio. En ambos casos, los elementos que estamos midiendo varían continuamente (sin "saltar" de un valor al siguiente) y las mediciones también (la aguja del velocímetro o la columna de mercurio pasan "suavemente", "sin saltos" de un valor a otro.) En un computador analógico pueden programarse operaciones (por ejemplo suma, resta, etc.) entre estas mediciones, siempre en base al mismo principio de representación.

La exactitud de estas mediciones y resultados depende de la precisión de los aparatos componentes.

El principio del computador digital es el de representar las magnitudes en cuestión por números o caracteres expresados directamente como combinación de dígitos (por ejemplo de ceros y unos en los equipos modernos) y las operaciones entre dichas magnitudes "reales" se reflejan en el equipo como operaciones entre dichos números.

El reloj digital representa un segundo mediante una cierta cantidad de pulsos eléctricos, digamos un millón. Cuando un contador detecta el millonésimo pulso, el medidor de tiempo se incrementa un segundo. La representación no es continua sino discreta, de a saltos (aunque sean muy pequeños).

Entre un pulso y otro el tiempo no transcurre para el reloj (aunque si en la realidad).

En otro campo, si una fotografía que tomamos con una máquina standard refleja en el vegativo las fluctuaciones continuas entre distintas luminosidades que serán en la realidad objeto de la foto, una imagen digitalizada (y de este tipo son las que transmiten los satélites) detecta y codifica la luz, en una serie de puntos de la superficie objeto, y transmite esa cadena de datos, que luego se reconstruyen de acuerdo al código preestablecido. Entre un punto y otro no hay medición alguna pese a que allí varía la luz. El secreto de la fidelidad de la foto, es en todo caso, el de disponer de muchos puntos muy próximos entre sí.

Si bien la computación analógica tiene vigencia, el avance tecnológico con la consecuente baja de costos y miniaturización de los equipos digitales, así como la mayor facilidad para encarar problemas generales de

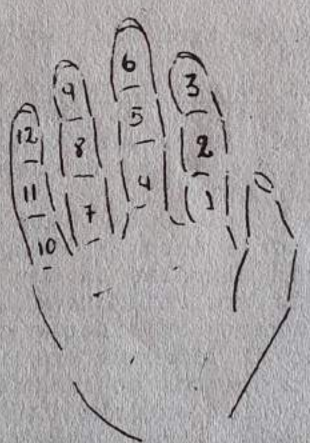
ordenamiento, almacenamiento y uso de información (no solo cálculos), unidos a la posibilidad de mejorar indefinidamente la precisión (por ejemplo, más pulsos por segundo o más puntos para la misma imagen) hacen que hoy se identifique computación con computación digital y con esta convención vamos a desarrollar el presente curso.

ENFOQUE HISTORICO DE LA COMPUTACION

ETAPA PRECONTEMPORANEA

¿Alguna vez nos detuvimos a pensar que las razones por las cuales el sistema decimal ha sido universalmente aceptado no son de índole matemático? Los 10 dedos de las manos han constituído el aparato primario de cálculo que empleó el hombre desde los tiempos prehistóricos. Valiéndose de los dedos es fácil contar hasta diez. Al llegar a 10, es decir, después de consumir todas las posibilidades de nuestro "aparato de cálculo" natural, lo lógico es considerar el nº 10 como unidad nueva, mayor (la unidad de orden siguiente). Diez decenas forman la unidad de 3er. orden y así sucesivamente. Por lo tanto, precisamente el cálculo a base de los dedos de la mano ha dado origen al sistema que ahora nos parece completamente natural. Sin embargo el sistema decimal de numeración tardó mucho en ocupar la posición dominante que tiene actualmente.

En distintos períodos históricos muchos pueblos emplearon sistemas de numeración distintos, por ejemplo, tuvo bastante difusión el sistema duodecimal. Su origen también está ligado al cálculo por los dedos: ya que cuatro dedos de la mano (salvo el pulgar) tienen 12 falanges en total, pasando el dedo pulgar por estas falanges se puede contar de 1 hasta 12.



Después se pasa a la unidad de orden siguiente, etc. Aún conservamos vestigios de este sistema cuando hablamos de "docena" - inclusive antes se hablaba de la "gruesa" (doce docenas) y de la "masa" (docena de gruesas).

En Inglaterra lo encontramos en el sistema de medidas (1 pie = 12 pulgadas) y en el sistema monetario (1 chelín = 12 peniques).

En Babilonia antigua, cuya cultura (incluyendo la matemática) era bastante elevada, existía un sistema sexagesimal muy complejo. Existen varias hipótesis a cerca del origen de este sistema. Una es que se produjo la fusión de 2 tribus una de las cuales usaba el sistema senario y la otra el decimal, surgiendo como compromiso entre los dos sistemas el sexagesimal.Otra hipótesis es que los babilonios consideraban el año compuesto de 360 días lo que se relacionaba de modo natural con el nº 60. Pero ambas Hipótesis rc parecen demasiado acertadas. Sobretodo la segunda ya que los conocimientos astronómicos de los antiguos habitantes eran muy elevados

y pareciera que el error de estimación de duración del año en 5 días es muy frande.

Este sistema se ha conservado aún en nuestros días ya que una hora es igual a 60 minutos, un minuto es igual a sesenta segundos, y un grado igual a sesenta minutos, 1' = 60".

Según Stanley explorador de Africa, varias tribus de esa zona empleaban el sistema quinario (base cinco) - Es evidente la relación de este sistema con la "máquina computadora" primaria, los dedos de una mano.

Los aztecas y los mayas usaban el sistema vigesimal, al igual que los celtas en el occidente de Europa. En este caso se supone también su origen es "anatómico", deriva de los diez dedos de las manos más los 10 dedos de los pies.

Algunos vestigios del sistema vigesimal de los celtas subsisten en el idioma francés por ej., ochenta se dice "quatre-vingt" o sea, "cuatro veces veinte". El franco, unidad monetaria francesa, consta de 20 "sous".

Todos los sistemas señalados anteriormente se basan en el mismo principio general, son posicionales.

Si se toma un número p, base del sistema de numeración, todo número N se representa como la combinación de potencias de p con coeficientes que toman valores de 0 a p-1, o sea en la forma

$$a_k p^k + a_{k-1} p^{k-1} + \ldots\ldots\ldots + a_1 p + a_0$$

Después, este número se denota abreviadamente

$$(a_k\ a_{k-1} \ldots\ldots\ldots\ldots a_1\ a_0)_p$$

En este caso el valor de cada cifra depende del lugar que ocupa. Por ejemplo, en el número $(222)_{10}$ el dos figura tres veces; pero el de la extrema derecha representa dos unidades, el del medio significa dos decenas y el otro, dos centenares.

Generalizando $(222)_p$, estos tres dos significarán, respectivamente los valores 2, 2p y $2p^2$.

Existen también sistemas no posicionales que se basan en otros principios. El ejemplo mas conocido son los números romanos. En este sistema se tiene una colección determinada de símbolos, unidad I, cinco V, diez X, cincuenta L, cien C, etc. y todo nº se representa como una combinación ellos. Por ej. $(88)_{10}$ se escribe LXXXVIII.

En este caso el significado de cada símbolo no depende del lugar que ocupa. La cifra puede aparecer 3 veces y siempre vale 10 unidades. La mayor ventaja de los sistemas posicionales es que permiten realizar fácilmente las operaciones aritméticas (multiplíquese, para comprobar, dos números de 3 cifras en sistema romano y en sistema decimal).

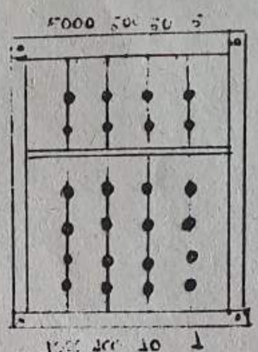
En definitiva, volviendo al tema que nos interesa en esta etapa que es el desarrollo de los instrumentos de cálculo, queda claro que desde 5000 años AC, el "aparato de cálculo" natural fué la mano del hombre, luego se extendió a contar con palos, piedras, rayas en las paredes, es decir con todo lo que sirviese para obtener un cálculo exacto.

Pero al desarrollarse el comercio, cuando los mercaderes viajaban comprando, vendiendo y haciendo trueques, en la medida en que el hombre acumulaba mayor número de posiciones y aumentaba la complejidad de la tarea de contar, los dedos no alcanzaban para obtener toda la información en su intercambio comercial.

Aparece entonces el contador de arena de los egipcios. Se trazaban surcos verticales en la arena, se tomaban piedritas y se establecía una correspondencia entre los objetos a contar y las piedras. Al completar 10 piedritas con el surco de las unidades, se cambiaban por una sola en el surco siguiente de la izquierda, en lo que fue uno de los primeros dispositivos mecánicos de calcular.

Es acá, donde se visualiza concfetamente el mecanismo de manejo del sistema posicional. Un progreso con respecto al contador de arena lo constituyó el ABACO. Este se inventa en China se cree que 3000 años AC o posiblemente antes. Era portátil, se lo podía manejar sobre una mesa e incluso en un barco.

No hay cambio en cuanto la filosofía, sigue basándose en el sistema posicional.



El ábaco consiste en una fila de varillas verticales y paralelas entre sí, y donde se deslizan pequeñas bolitas que están montadas sobre un armazón rectangular.

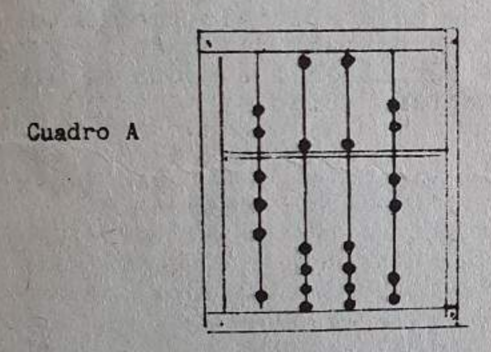
Las bolitas se dividen en dos secciones, por un travesaño que las separa. Las bolitas que están bajo el travesaño representan los siguientes valores: las encolumnadas en la primera varilla de la derecha, representan 1 cada una; las de la segunda de la derecha, 10 cada una; las de la tercera, 100; las de la cuarta, 1000 cada una.

Las bolitas que están sobre el travesaño, representan: las encolumnadas en la primera varilla de la derecha, 5 cada una; las bolitas de la segunda varilla, 50; las de la tercera 500 y las de la cuarta 5000.

La entrada de los números se realiza moviendo las bolitas hacia arriba (o hacia adelante, según la posición del ábaco).

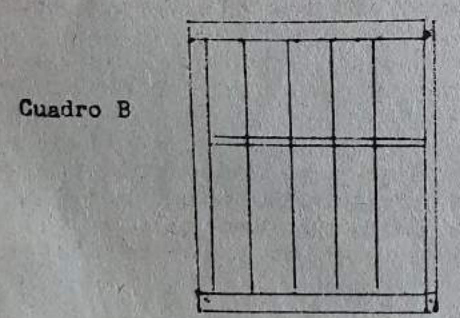
Si queremos, por ejemplo, representar la cifra 3552, adelantamos las bolitas correspondientes, y la lectura del ábaco será como se vé en el cuadro.

Cuadro A



Ahora llenen Uds. mismos con el lápiz, (cuadro B) las bolitas que habrían que adelantar para escribir la cifra 12342, pero, a fin de que se familiaricen con el ábaco, comiencen desde la derecha: primero ubiquen el 2, despúes el 40, luego el 300, el 2000 y el 10000.

Cuadro B



Aprendieron a entrar números y a leer en el ábaco. Escribieron una cifra que, con el contador de arena les hubiera requerido una tarea muy esforzada, llena de riesgos de error y poco práctica. Esto nos da una idea del salto fabuloso que significó este instrumento portátil y capaz de almacenar grandes cifras.

Pero no solo de almacenar, sino también de sumar, restar, multiplicar y dividir, todo ello a una velocidad asombrosa. Las sumas (adiciones) se hacen adelantando bolitas, las restas (sustracciones) invirtiendo el proceso de la adición. La multiplicación y la división se hacen con sucesivas adiciones y sustracciones.

Lo más notable del ábaco es que todavía hoy se emplea en bancos de China y parte de Japón ya que funciona con la misma rapidéz que las calculadoras mecánicas.

Los prÓximos instrumentos de cálculo, aparecen en Europa, en el Siglo XVI.

Los aportes que occidente hace en esta materia, se deben a los conocimientos que antes asimiló del Oriente.

Desde luego, el ábaco se impuso entre los romanos, desplazando a sus "calculi", unas piedrecitas que utilizaban para contar.

De ahí el origen de nuestra palabra "cálculo".

En el transcurso de la Edad Media, el ábaco se impuso en Europa. Pero en este período histórico, se da un proceso importante: los árabes transmiten desde España, sus conocimientos matemáticos basados en el sistema decimal, conocimiento que a su vez aprendieron de la India. También transmiten sus conocimientos de astronomía aprendidos de los chinos, y los conocimientos de filosofía, aprendidos de los griegos.

A través de un largo proceso, los europeos asimilan el sistema indo-arábigo de matemáticas, y también desarrollan la astronomía, una actividad que requiere mucho cálculo.

A fines del Siglo XVI aparece John Napier, escocés, un ilustre matemático dedicado a la astronomía, entre otras materias.

Napier sistematizó el uso del punto decimal, para separar la parte fraccionaria de la parte entera de los números. En 1549 comenzó a trabajar con logaritmos, a él debemos su invención.

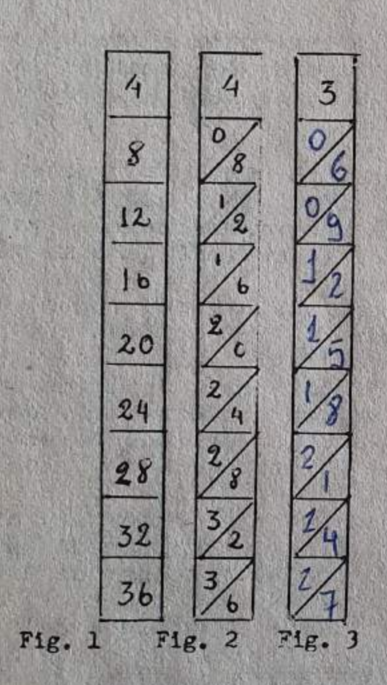


Fig. 1 Fig. 2 Fig. 3

En materia de instrumentos de cálculo, el gran aporte de Napier son sus famosas y populares "varillas", o también llamados "huesos de Napier". El asignó una varilla a cada número del 0 al 9, y escribió los múltiplos de cada uno en cuadraditos.
Por ejemplo la varilla del 4. Debajo suyo están: 8, 12, 16 etc., (fig 1)
Pero Napier separó la unidad de las decenas, de cada múltiplo, mediante una diagonal en cada cuadrado. (fig 2)
Llenen la varilla del 3 según el sistema de Napier. (fig. 3)

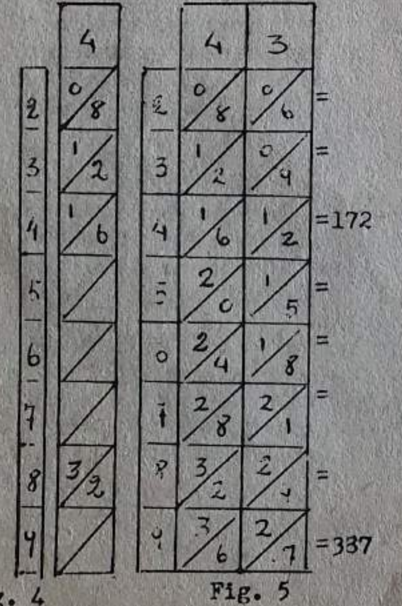


Fig. 4 Fig. 5

Si colocamos junto a la varilla del 4, otra varilla con los números corridos del 2 al 9, tenemos la tabla de multiplicar del 4 (fig. 4).
Si hacemos lo mismo con las varillas del 4 y del 3, podemos multiplicar, mediante simple suma, por 43. (fig. 5)
En las varillas, tenemos el valor de 9 x 4 y el valor de 9 x 3.

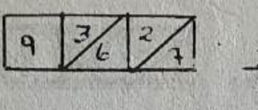


Fig. 6

27
36
387

Fig. 7

En la fig. 6 hemos aislado la lectura 9 x 43. Ahora sumamos como en la fig. 7 y tenemos el resultado de 9 x 43. Completen Uds. la tabla del 43. (fig. 5)

Los grandes descubrimientos geográficos del Siglo XVI, van permitiendo a Europa transitar, desde una economía de autosubsistencia, a otra de intercambio. O sea, se comienza a producir, no ya para satisfacer las necesidades de una comunidad, sino también para intercambiar productos propios con los de otras comunidades. Esto permitió que las economías europeas desarrollasen su artesanía (producción de muebles, relojes, herramientas de todo tipo, armas, ropa, etc.) una actividad de tipo familiar que se ubica en las ciudades que por entonces se llamaban "burgos".

Los "burgos" europeos comienzan a crecer, son asiento de actividades artesanales, comerciales, bancarias, políticas y administrativas, además de universitarias.

Blas Pascal nace en uno de esos grandes burgos de la Europa de comienzos del Siglo XVII: París. Su padre era un típico burgués ("Hombre de los Burgos"), perteneciente al régimen administrativo del reino, como recaudador de impuestos.

Esta era una tarea que se hacía cada vez más compleja, en una sociedad que también se tornaba cada vez más compleja.

Pascal revolucionó su época por sus aportes en el campo de la filosofía, pero también fue un genio matemático, y además revolucionó el campo de la computación.

A fin de aliviar las tareas del negocio de contabilidad de su padre, y aprovechando el desarrollo de la artesanía metalúrgica, que ya construía piezas con gran precisión, ideó y construyó un instrumento mecánico para sumar y restar.

Fue en el año 1644.

Su invento se basó en una idea simple: la operación de sumar realizada mediante el agregado de elementos unitarios, que en su caso eran sectores de la rotación de un cilindro. Pensó que asignando a la rotación completa de un cilindro el valor de 10 unidades, cada vez que el cilindro hacía un décimo de dicha rotación completa, a partir de una posición inicial dada, agregaba una unidad al valor inicial considerado.

Finalmente implementó esta idea con un aparato que consistía en la vinculación de varios cilindros por un ingenioso dispositivo gracias al cual, cuando un cilindro realizaba un giro completo, volviendo a colocarse en la posición cero. arrastraba mediante una pestaña al cilindro adyacente e impulsaba en este segundo cilindro una rotación unitaria por cada giro completo del primer cilindro. En 1666, Sir Samuel Moreland, perfeccionó la máquina de Pascal, logrando la multiplicación mediante técnicas de sumas acumulativas.

Hacia fines del Siglo XVII, en 1694, el alemán Gottfried W. Leibnitz termina la construcción de su "Contador Escalonado", una máquina más avanzada que la de Pascal, capaz de sumar, restar, multiplicar, dividir y extraer raices cuadradas. El proceso se realizaba por una serie de sumas repetidas, al igual que las actuales computadoras digitales.

Pero la mayor contribución de Leibnitz fue el concebir el sistema de numeración binaria que permite escribir todos los números y operar con ellos utilizando sólo dos símbolos: el cero (0) y el uno (1).

Por supuesto siguiendo las normas que ya vimos de cualquier sistema posicional donde $p = 2$ y los coeficientes a_i toman sólo dos valores (0 o 1), los números se escriben de la siguiente manera:

cero	0	tres	11	seis	110	nueve	1001
uno	1	cuatro	100	siete	111	diez	1010
dos	10	cinco	101	ocho	1000	once	1011

doce	1100	dieciseis	10000
trece	1101	diecisiete	10001
catorce	1110	treinta y dos	100000
quince	1111		

Las operaciones se efectúan con los mismos principios con los que se opera con números escritos en sistema decimal

```
   101 (cinco)            1100  (doce)
+   11 (tres)           -  101  (cinco)
  ----                  ------
  1000 (ocho)              111  (siete)


   1111   (quince)        (veintiocho)  11100 | 111  (siete)
   x 11   (tres)                        00000   100
   ----
   1111
  1111
 ------
 101101   (cuarenta y cinco)
```

Las tablas + y x son muy sencillas

+	0	1
0	0	1
1	1	10

x	0	1
0	0	0
1	0	1

La única dificultad de este sistema es la gran cantidad de ceros y unos que son necesarios para escribir los números.

Leibnitz también hace contribuciones desde el campo de la lógica. Su preocupación por llegar a construir una "máquina de razonamientos" lo condujo a continuar la lógica como un álgebra de posiciones, es decir, como un álgebra que opera no con números sino com proposiciones que pueden tener sólo dos valores (verdadero o falso). Fue Serge Boole, un matemático inglés, quien desarrolló estas ideas en su libro "Leyes del Pensamiento" (1854).

El Siglo XVIII no presenta avances notorios en el terreno de la computación. Pero este es un siglo muy rico en hechos históricos, que prepara futuros avances.

En el campo político, Europa reemplaza los sistemas monárquicos de gobierno por los democráticos; y en el campo económico reemplaza los sistemas artesanales de producción, por los industriales.

La revolución industrial se despliega en el Siglo XVIII con la introducción de una nueva fuente de energía: el vapor. El escocés Jaime Watt (nacido en 1736) concibe el principio de la máquina de vapor de doble efecto, que se aplicará para mover máquinas, barcos, carros y locomotoras.

Todos estos "inventos" se orientan a reemplazar el esfuerzo físico del ser humano, mientras que el desarrollo de las "máquinas de calcular" apuntan a eliminar el esfuerzo mental siendo ambas vertientes un aporte a incremetar la productividad del trabajo humano.

En el marco de avances tecnológicos, un ingenioso francés inventa el telar automático, que tendrá enormes repercusiones en el área de la computación, incluso hasta nuestros días.

En efecto, joseph Marie Jacquard era un ingeniero de la industria textil, que no pensaba en como mejorar las máquinas calculadoras sino la producción de ropa.

Preparó una cadena sin fin de tarjetas perforadas, para que girara pasando por las agujas de un telar especialmente diseñadao. Sólo cuando una aguja coincide con uno de los agujeros de la tarjeta, puede penetrar, e incorporar un hilo en el tejido. Cada cadena de tarjetas constituiría un verdadero programa para la máquina tejedora, la cual podía incorporar sobre un paño, toda suerte de figuras y diseños, según la ubicación de los agujeros en las tarjetas.

El telar automático de Jacquard se incorporó a la producción a comienzos de 1800, aunque su primer modelo fue concluído varios años antes.

En el año 1835, el matemático inglés Charles Babbage, imagina su "máquina analítica", primera computadora de la historia, utilizando la tecnología de la tejedora de Jacquard.

Babbage concibe una máquina capaz de realizar no un cálculo aislado sino una serie ordenada de cálculos que se realizan según una serie de órdenes - Utilizaba para ello dos conjuntos de tarjetas perforadas de Jacquard: uno con los datos (iniciales e intermedios) que constituía una unidad de almacenamiento; el otro conjuntos de tarjetas perforadas tenía la secuencia de operaciones, constituyendo una unidad de procesamiento de datos. (Ver en el Anexo mayor detalle acerca de Babbage y su invento).

Los émbolos pasaban a través de las perforaciones de las tarjetas y operaban los mecanismos para transferir los números del almacenamiento a la unidad de procesamiento de datos.

Así como el telar automático de Jacquard tejía flores y hojas, la Máquina Analítica de Babbage "tejía modelos algebraicos, de un modo también autocontrolado.

Los más notables de la Máquina Analítica es que contenía todos los rasgos escenciales que forman la computadora electrónica moderna.

Sin embargo, Charles Babbage fue un genio adelantado a su época. Preparó planes detallados para la concreción de su máquina, pero no existían las condiciones tecnológicas para consumarlos. Sus escritos se sumergieron en el olvido y un siglo después, en 1937, son redescubiertos, para servir de apoyo a nuevos adelantados.

La tecnología de los telares de Jacquard es retomado por otro pionero de la computación, el estadístico norteamericano Herman Hollerith.

Hollerith trabajó en el censo de los Estados Unidos de 1890. Diez años antes, se habían censado todos los habitantes de este país, y los administradores de la Oficina de Censos habían llegado a la conclusión de que, cuando las cifras estaban por fin clasificadas, no eran más utilizables. En 1887, ya próximos a la fecha del nuevo censo, se comenzaron a buscar procedimientos que permitieran manejar con mayor eficiencia, una masa de datos que iba a ser mucho mayor.

Entre otras alternativas, algunos propusieron reemplazar el viejo sistema de los libros, por tarjetas de colores y signos para facilitar el cálculo.

HOMBRE
Nombre ....
Edad ....
XXX ....

Hollerith recurrió a las tarjetas perforadas de Jacquard. Asignó una ficha a cada persona censada y perforó, a la derecha, las de los hombres y a la izquierda, las de las mujeres.

MUJER

Nombre............

Edad..............

XM ....

La perforación de la ficha, permite que una determinada varilla metálica se introduzca en un dispositivo que acciona un contador. Cada uno de los dos contadores -el de los hombres o el de las mujeres- va sumando las tarjetas correspondientes y almacenando las cifras.

Estos contadores fueron construídos según la tecnología ideada por Blas Pascal, para su máquina calculadora.

Luego Hollerith dio un paso más, utilizando la energía eléctrica para accionar los contadores. En lugar de varillas metálicas, por el agujero de la ficha pasa la electricidad, hacia uno de los dos contadores: el de los hombres, o el de las mujeres.

HOMBRE BLANCO	MUJER BLANCA
HOMBRE DE COLOR	MUJER DE COLOR

Nombre ....

Edad ....

Profesión ....

Pero el censo de los Estados Unidos discriminaba además de los hombres y las mujeres, a los blancos de los negros.

Hollerith diseñó una nueva tarjeta con cuatro casilleros: para los hombres blancos, los hombres negros, las mujeres blancas y las mujeres negras.

Según de quien se tratase, se le asignaba una tarjeta perforada en uno de los cuatro casilleros.

Y conectó cuatro contadores en su máquina.

El censo de 1890 ocupó la tercera parte del tiempo del de 1880, no obstante que, en los 10 años transcurridos, la población se incrementó considerablemente.

A una velocidad de "lectura" de 80 tarjetas por minuto, la máquina de Hollerith computó y clasificó a 62.622.250 habitantes.

Resumiendo, H. Hollerith integró dos tecnologías básicas: la de Jacquard y la de Pascal. La primera reemplaza al hombre en acciones concretas como tejer; la segunda, en cálculos mentales, como sumar. Esta nueva máquina, que además incorporó la energía eléctrica para su movimiento, reemplazó al hombre en la pesada y tediosa tarea de leer millones de fichas para sumarlas clasificadamente. A partir de una masa desordenada de datos produjo una información, todo ello a gran velocidad.

Hollerith, incorporando elementos ya inventados, los aplica a un uso hasta entonces desconocido: la manipulación de grandes volúmenes de datos.

Herman Hollerith fue uno de los socios fundadores de la International Bussiness Machine (I.B.M.).

CUADRO SINOPTICO HASTA EL SIGLO XIX			
FECHA	LUGAR	INVENTOR	FUNCIONES Y CARACTERISTICAS
5000 años A.C.		?	Uso de los dedos para sumar o restar.
3000 años A.C.	Medio Oriente	¿Egipcios?	Contador de arena. Permite sumar o restar y agrupar de a 10 unidades.
3000 años A.C.	Lejano Oriente	¿Chinos?	Abaco. Permite sumar, restar, multiplicar y dividir.
Siglo XVI	Escocia	John Napier	Varillas o huesecillos. Basado en el sistema decimal y uso de Número Arábigo. Permite multiplicar con varios dígitos.
Siglo XVII	Francia	Blas Pascal	Máquina de sumar y restar. Primer sistema mecánico.
Siglo XVII	Inglaterra	Samuel Moreland	Perfecciona a Pascal. Su máquina suma, resta y multiplica.
Siglo XVII	Alemania	G. Leibnitz	Contador escalonado. Puede sumar, restar multiplicar, dividir y extraer raíces cuadradas. Basado en sistema binario.
Siglo XVIII	Francia	J.M. Jacquard	Aporta involuntariamente a la computación con su telar automático, a base de tarjetas perforadas.
Siglo XIX	Inglaterra	C. Babbage	Máquina analítica. Utiliza tarjetas perforadas. Contiene todas las características de las computadoras electrónicas. No se construye.
Siglo XIX	E.E.U.U.	H. Hollerith	Combinación de la técnica de Jacquard y la de Pascal. Incorpora electricidad. Lee, cuenta y clasifica tarjetas del censo.

ETAPA CONTEMPORANEA

- Introducción

Estamos ubicados ahora en los años 30 del presente siglo. Es esta una década conflictiva, se abre con una tremenda crisis económica mundial que servirá de caldo de cultivo a la segunda guerra mundial que estalla al final del período involucrando a todas las potencias industriales europeas.

Los Estados Unidos recién se incluye en el conflicto en 1942 y prácticamente no sufren los efectos de la guerra, por lo que inauguran un período de supremacía industrial y militar.

Esta situación explica, por un lado, que las principales motivaciones para el desarrollo de máquinas de cómputo hayan sido las aplicaciones bélicas (balística, cohetería, energía nuclear, decodificación de mensajes), y por el otro, que los adelantos en esta materia provengan escencialmente de los E.E.U.U.

- El ajuste entre la ciencia y la técnica

Por caminos ajenos a la computación, la tecnología de la que carecieron los precursores se fue desarrollando. Es en una rama de la ingeniería eléctrica, la ingeniería de comunicaciones, donde se produce la conquista decisiva: el perfeccionamiento en el uso de los controles automáticos.

En particular la telefonía, al transformarse en automática impone el uso de los "relais" o relés (dispositivos electromecánicos) que son vinculados entre si por circuitos que comienzan a ser diseñados empíricamente de acuerdo a las necesidades y que llegan a ser un punto clave en el desarrollo de la moderna computación.

Ahora podían replantearse las ideas de Babbage en un nuevo contexto.

Esto es lo que hizo Aiken de la Universidad de Harvard en 1937, quien reexaminó los escritos olvidados del inglés, y se volcó entre 1938 y 1942 a construir la primera gran computadora electromecánica, la MARK I.

Esta precursora tenía 15m. de largo por 2,40m de alto. Era accionada y controlada por una cinta de papel perforado (allí residía el "programa" a ejecutar) y realizaba las cuatro operaciones elementales y búsquedas en tablas. Los datos se ingresaban en tarjetas perforadas y la salida (los resultados) se registraba en tarjetas o a través de máquinas de escribir eléctricas. Lo mismo que sus predecesores desde Pascal hasta Hollerith, el almacenamiento de números se efectuaba en registros mecánicos.

A la MARK I le siguió la MARK II, hacia fines de la guerra y con aplicaciones militares (cálculo de funciones trigonométricas, objetivos ópticos y trayectoria de cohetes). Era más rápida gracias al reemplazo del sistema electromecánico por otro electromagnético.

En 1946 se construyó la ENIAC en la Universidad de Pensylvania. Las siglas significan "Calculador e integrador numérico electrónico" y era la primer computadora realmente electrónica.

El programa que efectuaba el control de las operaciones no era ingresado externamente sino que se ejecutaba a través de cables integrados al propio cuerpo de la máquina.

La ENIAC (construída por Eckert y Mauchly) era 100 veces más veloz que sus predecesoras electromecánicas, y se estimó entonces que los cálculos de física nuclear que realizaba en 2 horas hubieran demandado un año de trabajo a 100 ingenieros.

Sin embargo todas estas máquinas eran extremadamente rígidas debido a la dificultad de cambiar un programa por otro. "Programar" la ENIAC suponía cambiar la forma del cableado interno para que la computadora ejecutase otra secuencia de órdenes.

Una idea que ya estaba en los escritos de Babbage y en el diseño que el matemático inglés Alan Turing publicó en 1936 (la máquina de Turing que se verá mas adelante) significó un nuevo salto conceptual: almacenar el programa o secuencia de instrucciones en la memoria del computador como si fueran otros tantos datos.

Este concepto es esencial en la llamada "máquina de Von Neumann" (por ser este el que logró efectivizarlo primero) y sigue estando vigente hasta la actualidad.

Retrocedamos unos años para analizar con más detalle algunas ideas rectoras del diseño de las modernas computadoras.

UNA REVOLUCION EN EL PENSAMIENTO

A principios de Siglo, el eminente matemático David Hilbert, a partir del planteo de una lista de problemas no resueltos en la matemática contemporánea, propuso como desafío la elaboración de un método general para determinar si un teorema es o no verdadero.

En 1936 otro matemático, el joven Alan Turing llegó a la conclusión de que el problema de Hilbert era imposible de resolver.

La publicación en la que Turing anunció su resultado ha tenido una significación y trascendencia que rebasan con mucho el problema a que inmediatamente se dirigía. Al atacar el problema de Hilbert, Turing se vio forzado a plantearse cómo dar al concepto de método una definición precisa. A partir de la idea intuitiva de que un método es un algoritmo -un procedimiento que puede ser ejecutado mecánicamente sin intervención creativa alguna- Turing hizo ver cómo esta idea puede refinarse y convertirse en un modelo detallado del proceso de computación, en el cual un algoritmo cualquiera es descompuesto en una secuencia de pasos atómicos simples. El modelo computacional resultante es una construcción lógica conocida por máquina de Turing.

La forma más sencilla de describir la máquina de Turing es mediante elementos mecánicos, tales como ruedas, cintas perforadas y un sensor capaz de desplazarse hacia adelante y atrás, a lo largo de la cinta. Si bien toda esta maquinaria no es imprescindible, pues en su nivel fundamental la máquina de Turing no es sino la concreción material de un método de razonamiento matemático, la eliminación total de la analogía mecánica podría inducir a confusiones. Tal analogía le resultaba sugerente al propio Turing, que fue uno de los pioneros en el desarrollo del ordenador digital. Además, los méritos de la máquina de Turing en tanto que instrumento conceptual son hoy proclamados por las ciencias de cómputo con, por lo menos, tanta fuerza como pueda hacerlo la lógica. En las ciencias de cómputo su importancia es fundamental: si a la máquina de Turing se le da tiempo suficiente, que será mucho, pero finito, podrá llevar a cabo cualquier cómputo que pueda realizar un moderno ordenador digital, por muy potente que sea.

Esta capacidad universal de la máquina de Turing no implica que pueda ser utilizada de modo práctico como ordenador. Los ordenadores reales pueden funcionar a velocidad enormemente mayor que la máquina de Turing, porque al diseñarlos se sacrifica deliberadamente la claridad de funcionamiento en aras de mayor eficiencia y velocidad. No obstante, en el estudio teórico de la capacidad última que para la resolución de problemas pueda tener el ordenador real, la máquina de Turing se ha hecho indispensable.

En el anexo respectivo se describe en detalle la máquina de Turing y se propone a cada lector que se construya una y la haga funcionar de inmediato.

Recomendamos asimismo analizar el concepto allí expuesto de Máquina Universal de Turing ya que allí está la base de la idea del programa almacenado "en memoria" como si fuera otro conjunto de datos que ya comentamos al presentar las ideas de Von Neuman.

Una consecuencia interesante de las máquinas de Turing se refiere al límite de lo calculable. Sugerimos antes de seguir, meditar esta pregunta ¿Considera el lector que con suficiente tiempo y poderosas computadoras cualquier problema o cálculo pueden resolverse?

Si aceptamos que para cada cómputo que deseamos realizar hay una máquina de Turing que lo ejecuta a condición de tener suficiente tiempo, la pregunta anterior podría formularse ¿Habrá sufiecientes máquinas de Turing (o suficientes "programas") como para resolver todos los cálculos posibles?

A lo largo del curso volveremos sobre este interrogante para intentar darle respuesta.

EL ESQUEMA DE VON NEUMANN

Si

John Von Neumann se dedicaba a fines de los años 30 al estudio de complicados problemas de hidrodinámica, cuya resolución no parecía posible sin el auxilio de una computadora. Conocía por otra parte los escritos de Turing y los del norteamericano Emil Post que, coincidentemente desarrollaron el fundamento lógico de la computadora con un programa almacenado.

Un encuentro fortuito con uno de los científicos que se encontraba desarrollando la ENIAC posibilitó a Von Neuman una relación asidua con dicho grupo al que en sucesivas reuniones le planteó las características que a su juicio, debería poseer la computadora capaz de resolver las dificultades que el tenía entre manos.

Para Von Neumann ésta debía ser una máquina general, independiente del problema a resolver, con un órgano llamado memoria donde se almacenen tanto las instrucciones detalladas para hallar solución al problema que se esté "procesando", como los datos y resultados numéricos del mismo.

A la vez otro órgano (una "Central de Control") debía controlar la ejecución en orden de las intrucciones sin importarle que es lo que querían resolver esas instrucciones (esto le daba un carácter flexible, posibilitando que distintos problemas se resuelvan con el solo recurso de cambiar la secuencia de instrucciones en la memoria).

Las cuatro operaciones aritméticas las realizaría una parte especializada en esta tarea llamada unidad aritmética y, finalmente, unidades de entrada y de salida se encargarían de obtener los datos y comunicar los resultados respectivamente.

El esquema incluía el modo de operación en serie, es decir, analizando y ejecutando una instrucción por vez, (la ENIAC por ejemplo, realizaba varias tareas simultáneas).Vale la pena comentar que 30 años de avances espectaculares en la tecnología computacional convirtieron a este modo de operar en un "cuello de botella" que frena el aumento de velocidad de proceso por lo que las futuras generaciones de máquinas adoptarán sin duda el esquema de proceso en paralelo de varias instrucciones a la vez.

En el apéndice sobre Von Neumann, el lector hallará mayores detalles acerca de la "arquitectura" propuesta por este científico para las modernas computadoras.

DE VON NEUMANN A LA ACTUALIDAD

Si bien las ideas de Von Neumann tuvieron aceptación, la ENIAC ya se encontraba en un avanzado estado de desarrollo por lo que no resultaba factible adaptarla al nuevo esquema.

El mismo equipo conducido por Eckert y Mauchly construyó entonces la EDVAC concluída en 1952, según los principios expuestos por Von Neumann y ya delineados un siglo atrás por Babbage.

La EDVAC, mas poderosa que la ENIAC, utilizaba la notación del sistema numérico binario.

En la década del 50 se inicia la etapa de la fabricación de computadoras para ser ofrecidas al mercado.

Esta etapa se inicia con la UNIVAC I (computadora automática universal), fabricada en 1951 por la Remington Rand Corporation, que fue utilizada por la Oficina de Censos de los Estados Unidos primero y para el procesamiento de datos comerciales, después.

La segunda computadora comercial fue la CRC 102, fabricada en 1952 por la Computer Research Corporation (NCR 102).

También en 1952 I.B.M. sacó al mercado la I.B.M. 701. Luego, en 1953, I.B.M. 702 e I.B.M. 650.

La producción, comercialización y uso de las computadoras recibe un impulso decisivo con el invento del transistor, ocurrido en 1948 y aplicado a lás máquinas calculadoras a partir de 1959.

La computadora transistorizada es más pequeña y más confiable que las grandes moles hechas con tubos al vacío.

Al comienzo de 1960, los fabricantes de transistores lograron colocar circuitos eléctricos completos sobre las superficie de una pequeña oblea de sílice (chips), llamados circuitos integrados.

En noviembre de 1970 se produce un salto en el desarrollo electrónico al producirse el primer microprocesador, es decir la computadora completa dentro de un "chip". Este pequeño dispositivo había así reemplazado la totalidad de los 18.800 tubos al vacío de la ENIAC.

Así es como, gracias a esta disminución de tamaño, peso y costo, hoy hay computadoras minúsculas que controlan lavarropas, despertadores, televisores, etc. y se plantean cada vez mas viejos y nuevos problemas que pueden resolverse mediante la herramienta de la computación.

Paralelamente a este pantallazo del desarrollo electrónico que permite máquinas cada vez mas veloces, potentes y versátiles, se va transformando la forma en que el hombre se acerca y se comunica con la máquina. Desde aquellos monstruos de los años 50 sólo programables por unos pocos ingenieros o matemáticos, hasta la computadora hogareña hay un salto quizás no tan espectacular como en la electrónica, pero sin duda fundamental.

EVOLUCION DEL SOFTWARE

Las primeras computadoras automáticas electrónicas fueron todas únicas y construídas en un laboratorio experimental. Fueron tan voluminosas que dificilmente se las podía mover y además, requerían un mantenimiento tan exclusivo que era bastante natural que el lugar donde la gente trataba de usar la máquina era el mismo laboratorio donde la máquina había sido desarrollada. Los programas que se hacían tenían sólo un significado local y con una utilidad acotada en el tiempo.

En general, eran poco confiables, muy lentas y su memoria era pequeña. Además había que instruirlas con códigos complicados y extraños. El que se encargaba de esto, en realidad, vivía un desafío permanente similar a armar rompecabezas y descubrir "tretas" inteligentes.

Pero en las décadas siguientes surgen máquinas poderosas en varios órdenes de magnitud, y sobreviene la "crisis de la programación" ya que dichas máquinas eran bastante más difíciles de manejar si se seguían pensando las cosas a nivel de la forma en que estaban construídos sus circuitos, diseñados sus almacenamientos, etc.

Como el poder de las máquinas disponibles creció en un factor de más de 100, también la ambición de la sociedad para aplicar esas máquinas creció en proporción.

Es así que se va también transitando una historia en el modo en que los hombres instruyeron las máquinas.

Como dijimos, inicialmente, se utilizó un "lenguaje de comunicación" con ellas que estaba directamente relacionado con su circuitería. Más adelante se conciben niveles más cercanos al lenguaje comprensible por el hombre o sea se avanza en la técnica de codificación. Luego se intenta un paso adelante donde el lenguaje de comunicación traiga implícito una metodología de pensamiento que permita encarar los problemas de forma más cercana a los modelos de la realidad.

En la actualidad se desarrollan preferentemente lenguajes orientados a captar y expresar el problema a solucionar, "obligando" a la máquina a entender las instrucciones a ese nivel (en realidad se realizan traducciones dentro del computador, pero eso no lo ve el usuario). El objetivo al que se apunta cada vez más es que las máquinas comprendan el "lenguaje natural" es decir, el lenguaje normal usado entre las personas con todos sus giros y sobreentendidos.

LA COMPUTACION HOY (Lo que se puede y no se puede esperar de las máquinas)

La computación ha producido grandes transformaciones en nuestra época y las continuará produciendo.

En el comercio, en la producción, en la política, en la investigación científica, en la guerra..., en cualquier actividad que pensemos, el manejo de la información da ventajas.

No basta que una decisión sea correcta, debe ser también oportuna, aplicada en el momento preciso. Una información oportuna, permite tomar decisiones oportunas.

Hoy día se producen grandes cantidades de datos, que deben ser almacenados y procesados con rapidez.

Además, con el desarrollo de las telecomunicaciones, podemos anular las distancias y conocer al instante lo que ocurre en los mercados, en los gobiernos o en las fábricas de cualquier lugar del mundo (alzas o bajas de monedas, decretos y leyes que afecten el intercambio mundial, lanzamiento de nuevos productos, ect.).

La computación, entonces, no se limita al cálculo rápido, nos permite también manejar información.

LOS MITOS DE LA COMPUTACION

Aplicaciones como los sistemas expertos que juegan al ajedrez o "razonan" acerca de temas sumamente especializados (diagnóstico médico, prospección geológica, etc.) sacando conclusiones análogas o superiores a un conocedor individual del tema y la aparición en escala de robots industriales, parecen avalar imágenes fantásticas de máquinas inteligentes, con creatividad, sentimientos e impulsos autónomos y aún contrapuestos a los del hombre. Nada más falso sin embargo. Todas las acciones de las computadoras son rutinarias en algún sentido. Incluso cuando realizan "deducciones", lo hacen a partir de una serie de pasos preestablecidos que intentan simular algunos aspectos del razonamiento humano. En definitiva son los hombres los que construyen modelos conceptuales de la realidad, y alimentan con los mismos a las máquinas,para luego observar las complejísimas manipulaciones que de los datos del modelo hacen las computadoras y luego sacar sus propias conclusiones.

El sentido de los datos y conclusiones se los pone el hombre; para el significan cosas concretas (nº de legajo y sueldos o trayectorias de dos cuerpos en el espacio) mientras que para la máquina son sólo cadenas de caracteres que maneja de acuerdo con ciertas reglas.

El hecho de que un sistema experto en diagnóstico de enfermedades infecciosas supere a un médico medio, sólo refleja el hecho de que la máquina se guía por reglas estipuladas por la experiencia colectiva de los mejores especialistas y tiende a reflejar el conocimiento social, acumulado, más que el de un individuo.

En todo caso, descartando las no siempre desinteresadas imágenes de las máquinas que dominan al hombre, muchos expertos comienzan a cuestionarse como controlar las posibilidades que semejante "simulación de inteligencia" pone en manos de los dueños de la tecnología, los que definen los modelos, captan informaciones sin límites de fronteras y manejan los resultados. Precisamente en nuestro continente latinoamericano ha surgido,en recientes encuentros de destacadas personalidades,la preocupación por el tema "soberanía e informática" como dos aspectos indisolubles,cuya criticidad se manifiesta en el llamado "flujo de datos trans-

frontera" que permite con frecuencia a poderosas empresas transnacionales disponer de mejor información sobre un país que su propio gobierno.

El otro mito emparentado al anterior es que las máquinas producen desempleo al reemplazar irreversiblemente el trabajo humano por el propio. Esto no es algo intrínseco a las computadoras y en todo caso debería ser resuelto por la organización social. El hombre es irremplazable en las actividades que requieren creatividad, en la definición de criterios de acción y en temas de decisión por voluntad, gusto o inspiración. Bien empleadas las máquinas reemplazan al hombre en las tareas rutinarias, dejándole más tiempo libre para el desarrollo de sus potencialidades.

SINTESIS DE LA UNIDAD I

La historia de la tecnología nos muestra la tendencia del hombre a generar el máximo de trabajo con el mínimo de esfuerzo. Cosa que va logrando con el recurso de su ingenio, cuya expresión más rica es, quizás, el instrumento.

El instrumento facilita las tareas del hombre y, en ciertos casos, llega a reemplazarlo.

Por ejemplo: con el invento del remo, el hombre multiplica su fuerza muscular y aumenta sus posibilidades de navegación. Luego, con la vela, introduce un nuevo instrumento que le permite librarse de la dura tarea de remar. Finalmente, con el motor, rompe su dependencia del clima y de los vientos, satisfaciendo mejor esa tendencia de que "algún otro" haga las cosas por él, y mejor.

Esta misma tendencia quedó demostrada en nuestra historia de la computación. El paso del contador de arena al ábaco significó facilitar enormemente las tareas de cálculo aritmético; con la calculadora de Pascal el hombre es directamente reemplazado en las tareas de sumar y restar, y luego con el contador de Leibnitz, en las tareas de multiplicar, dividir y extraer raíces. En 1890, la máquina de Hollerith ejecutó las tareas de leer, contar y clasificar las tarjetas de los censados. En fin, la ENIAC (1946), hizo en dos horas un trabajo de cálculo equivalente al de 100 ingenieros durante un año.

Sin duda el desarrollo tecnológico desplegado en los últimos 50 años no tiene parangón en toda la historia de la humanidad, siendo la computación uno de sus exponentes más avanzados.

Es que la computadora, como ningún otro instrumento, presenta posibilidades de reemplazar al hombre en la ejecución de tareas, permitiéndole ahorrar y disponer no ya de su energía física sino mental. Dicho en pocas palabras: con la ayuda de otros instrumentos, el hombre ejecuta sus tareas; con la computadora, da órdenes para su ejecución.

Es por eso que el objetivo no es ya saber cómo manejar una máquina, sino cómo darle órdenes.

ANEXO: Sistema Binario

CONVERSION DE NUMEROS

Desde la escuela primaria nos acostumbramos a pensar y expresar los números en la notación decimal. Tan grande es esta costumbre que resulta muy difícil comprender una cantidad si la vemos expresada en otro sistema numérico. Por ejemplo, si vemos una propaganda de una agencia de viajes: "VIAJE A EUROPA POR SOLO $ 1000", seguramente nos entusiasmaremos mucho. Si entramos a la agencia a consultar las condiciones del viaje, y nos enteramos que $ 1000 está expresado en la base 21, nuestra excitación desaparecerá rapidamente. Por otra parte, para saber si $ $(1000)_{21}$ es o no un buen precio debemos ser capaces de saber convertir este número a la base diez y comparar el precio con el de otras agencias. (Ejercicio: Averigue cuánto vale el viaje).

El ejemplo anterior, aunque un tanto descabellado, nos revela nuestra dependencia del sistema decimal. Por otra parte, por motivos que ya vamos a comentar, las actuales computadoras digitales son"máquinas binarias" y no decimales. De aquí nuestro interés por entender el sistema binario y por aprender a convertir números del sistema decimal al binario y viceversa.

También nos interesan otros sistemas de números. En particular el sistema octal (expresar números en base 8) y el sistema hexadecimal (base 16), nos permitirán expresar grandes números binarios en forma sencilla y más compacta.

En esta sección describiremos métodos para convertir un número decimal a binario, un número binario a decimal, un número octal a binario y un número hexadecimal a binario.

Conversión de un número binario a decimal

Utilizando el método mediante el cual hemos definido un sistema de números posicional (ver texto principal), podemos expresar un número en una base B como un número decimal D, aplicando la siguiente fórmula:

$$D = \sum_{i=-M}^{N} d_i B^i = d_{-M}B^{-M} + \ldots\ldots + d_{-2}B^{-2} + d_{-1}B^{-1} + d_0B^0 + d_1B^1 + \ldots\ldots$$

$$+ d_NB^N = d_NB^N + \ldots\ldots + d_1B^1 + d_0B^0 + d_{-1}B^{-1} + \ldots + d_{-M}B^{-M} \quad (1)$$

Por ejemplo:

$$(1101{,}011)_2 = 1x2^3 + 1x2^2 + 0x2^1 + 1x2^0 + 0x2^{-1} + 1x2^{-2} + 1x2^{-3} =$$

$$= 8 + 4 + 1 + 0{,}25 + 0{,}125 = 13{,}375$$

En este caso la base B = 2, los límites de la suma son $-M = -3$ y $N = 3$. Obviamente los dígitos d_i son 0 y 1 y el desarrollo de la suma se hizo de 3 a -3 (y no como en la fórmula (1)) por comodidad.

Ejercicios: Convertir a decimal los siguientes números binarios

a) 1011,1; b) 0,1011; c) 1101,001; d) 111,101; e) 1000,0011

Conversión de un número decimal a binario

El método consiste en dividir repetidamente el número por 2. En cada paso del procedimiento obtenemos un cociente y un resto de la división.

El cociente lo volvemos a dividir por 2. El número binario se obtiene escribiendo los restos de las divisiones sucesivas en el orden inverso del que aparecieron.

Por ejemplo, deseamos saber como se expresa el número 23 en base 2. Aplicando el método:

```
23 |_2_
 1   11 |_2_
      1    5 |_2_
            1    2 |_2_
                  0    1 |_2_
                        1    0
```

de este modo $(23)_2 = (10111)_2$

Este método es útil para convertir números enteros de la base 10 a la base 2. Para el caso más general de convertir un número racional, debemos dar un procedimiento para convertir fracciones decimales a la base 2. En este punto hay un problema. No todas las fracciones decimales se pueden expresar en la base 2 con un número finito de dígitos. Por lo tanto, debemos conformarnos con expresarlas sólo aproximadamente. Cuando convertimos una fracción decimal debemos decir con cuantos dígitos de exactitud lo hacemos.

El método de conversión consiste en multiplicar repetidamente la fracción decimal por 2. En cada paso del procedimiento obtendremos una parte entera y una parte fraccionaria del producto. La parte fraccionaria la volvemos a multiplicar por 2 y repetimos este paso tantas veces como dígitos exactos deseemos obtener. Escribiendo las partes enteras obtenidas en el proceso tenemos la fracción decimal expresando en la base 2.

Por ejemplo, deseamos escribir el número 0,75 en la base 2. Aplicando el método

paso 1.- $0{,}75 \times 2 = \underline{1}{,}50$

paso 2.- $0{,}50 \times 2 = \underline{1}{,}00$

así $(0{,}75)_{10} = (0{,}11)_2$

De esta manera el número $(23{,}75)_{10} = (10111{,}11)_2$

Ejercicios: Convertir a la base 2 los siguientes números decimales
a) 16; b) 639; c) 0,329 con cuatro dígitos exactos; d) 67,50
e) 127,25; f) 327; g) 96

Conversión de un número binario a octal y hexadecimal

Hemos dicho que el sistema octal y hexadecimal son de gran utilidad en la programación de computadoras. Esto se debe a que permiten expresar grandes números binarios en forma compacta.

En el sistema octal (el sistema de base 8) sólo existen los dígitos 0, 1, 2, 3, 4, 5, 6, 7. En el sistema hexadecimal (de base 16) los dígitos son 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, D, E, F. Las letras introducidas representan los números decimales 10, 11, 12, ..., 15. Sabemos que $2^3 = 8$. En consecuencia, cualquier grupo de tres dígitos binarios puede representarse por un único dígito octal. El procedimiento a seguir para convertir un número binario a la base 8, consiste en reemplazar ordenadamente grupos de 3 dígitos binarios por uno octal.

Ejemplo: Obtener la representación octal de $(1101{,}011)_2$

$$(1101{,}011)_2 = (001101{,}011)_2$$

agrupando de a tres dígitos a la izquierda y a la derecha de la coma, obtenemos: $\underbrace{001}_{1}\ \underbrace{101}_{5}\ ,\ \underbrace{011}_{3}$, es decir $(1101{,}011)_2 = (15{,}3)_8$

Utilizando el mismo argumento, si $2^4 = 16$, la conversión de un número binario a hexadecimal se realiza agrupando de a 4 dígitos. De este modo

$$(1101,011)_2 = (\underbrace{1101}_{D},\underbrace{0110}_{6})_2 = (D,6)_{16}$$

En general para transformar un número binario a uno en base K que sea potencia de 2 ($K = 2^n$, $n > 0$), el procedimiento consiste en agrupar ordenadamente los dígitos binarios en grupos de tamaño K y reemplazarlos por dígitos de la base.

Ejercicios: a) Convertir a octal y hexadecimal los siguientes números binarios $(100110,011)_2$, $(10101,0101)_2$, $(1111,111)_2$

b) Convertir a decimal los siguientes números: $(321)_8$, $(2F)_{16}$, $(634,21)_8$, $(333,33)_{16}$

c) Convertir los siguientes números decimales a las bases indicadas: 16 a la base 8; 124 a la base 16; 341,6 a la base 8; 2,321 a la base 16; 741 a la base 4.

OPERACIONES CON NUMEROS BINARIOS

Suma binaria: En la operación de adición entre números binarios existen cuatro posibles situaciones

1) $0 + 0$ 2) $1 + 0$ 3) $0 + 1$ 4) $1 + 1$

Las soluciones para las tres primeras situaciones son bastante evidentes, 0+0=0, 1+0=1, 0+1=1, pero la cuarta requiere de alguna explicación.
Razonemos por analogía con el sistema decimal; en este sistema contamos: 1,2,3,4,5,6,7,8,9,10,etc. ¿Que ocurre cuando pasamos de 9 a 10?, 9 es el mayor número que se puede representar en una dada posición. Si sumamos 1 a 9, debemos poner0 en la primer posición y acarrear 1 a la segunda. Simbólicamente

$$\begin{array}{r} 9 \\ +\ 1 \\ \hline 10 \end{array}$$

Exactamente lo mismo ocurre en el sistema binario. Puesto que 1 es el mayor dígito del sistema,

$$\begin{array}{r} 1 \\ +\ 1 \\ \hline 10 \end{array}$$

Resumiendo, la operación de adición en el sistema binario se efectúa de acuerdo a las reglas expresadas en la siguiente tabla:

+	0	1
0	0	1
1	1	10

Ahora que hemos examinado las 4 reglas básicas de la suma, debemos decir como efectuar la adición de números de varios dígitos. El algoritmo que debemos seguir es el mismo que usamos en el sistema decimal: sumar dígito a dígito de derecha a izquierda, efectuando el acarreo de un 1 cuando ocurra la situación 4) que hemos descripto antes.

Ejemplos : a) sumar los números 11010 + 11

```
  (1)
 11010
+00011
------
 11101
```

b) sumar 101 + 111

```
 (111)
  101
 +111
 ----
 1100
```

Ejercicios : Efectuar las siguientes sumas de números binarios :

a) 1101 + 1011
b) 1010 + 101
c) 101,11 + 110
d) 111 + 111
e) 0,101 + 10,11

Multiplicación binaria : El producto entre dos números binarios no ofrece mayores dificultades. Las reglas básicas quedan expresadas en la siguiente tabla

X	0	1
0	0	0
1	0	1

El producto de dos números binarios de varios dígitos se efectúa siguiendo el mismo procedimiento que para el caso de números decimales.

Ejemplo : a) multiplicar 101 x 10

```
  101
 x 10
 ----
  000
 101
 ----
 1010
```

b) multiplicar 1011 x 101

```
   1011
  x 101
  -----
   1011
  0000
 1011
 ------
 110111
```

Ejercicios : Efectuar las siguientes multiplicaciones entre números binarios

a) 1010 x 10
b) 111 x 11
c) 10.01 x 11
d) 0.01 x 1,1
e) 1100 x 1001,011

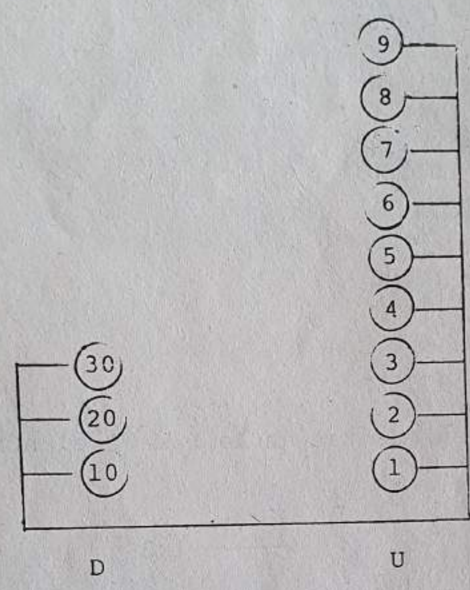
EL SISTEMA BINARIO Y LAS COMPUTADORAS

Aunque nos resulta completamente natural y eficiente cuando operamos con números, el sistema decimal no es conveniente para ser usado en una computadora digital. Su utilización requeriría del computador la capacidad de distinguir entre diez diferentes símbolos o configuraciones. El equipamiento necesario para dar a una máquina esta capacidad, la transformaría en un artefacto antieconómico y poco práctico.

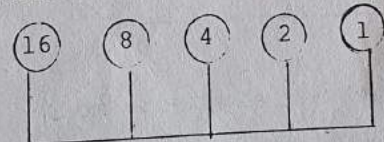
Ahora veremos como el uso de un sistema binario puede reducir el equipamiento necesario en un computador.

Supongamos que buscamos representar los números desde 0 hasta 31 por medio de un sistema de lámparas. Cada lámpara puede estar encendida (ON) o apagada (OFF).

En un sistema "decimal" las lámparas estarán organizadas en dos hileras. Si observamos el sistema en un momento dado, a los sumo una lámpara de cada hilera estará encendida.



Sumando los números de las lámparas encendidas, sabemos el número que el sistema "almacena" en ese instante. Así, si todas las lámparas estan apagadas tenemos el cero; si estan encendidas 10 y 5 tenemos 15, etc. Para representar los números entre 0 y 31 son necesarias doce lámparas.
El siguiente esquema muestra un sistema "binario":



Ahora tenemos cinco hileras, cada una de una lámpara. El sitema también es capaz de representar los números entre 0 y 31, pero sólo utiliza 5 lámparas.

En este sistema las lámparas representan los números 1,2,4,8 y 16, esto es, las cinco primeras potencias del número dos.

Cambiando el sistema "decimal" por el "binario" obtenemos dos importantes resultados, en especial, para las modernas computadoras digitales:

* El número de elementos (lámparas) se ve notablemente reducido.
* La naturaleza (encendido-apagado, ON-OFF) de los circuitos eléctricos queda completamente satisfecha.

VALORES NUMERICOS

En el ejemplo recién discutido, cada lámpara se encontraba en uno de dos estados :ON u OFF. Estos dos estados son usados para representar los símbolos del sistema de numeración binario: 0 y 1.

Una lámpara en ON equivale a 1; una lámpara en OFF equivale a 0.

También hemos notado que la posición de cada lámpara en el grupo tiene significado:

la posición de la derecha tiene valor 1, la siguiente 2, la tercera 4, etc.

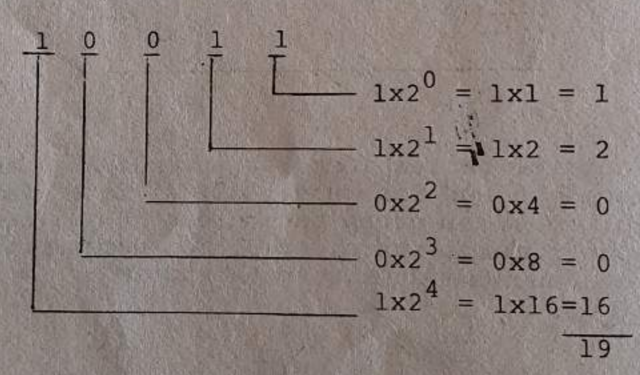
Sumando las posiciones de las lámparas encendidas, construíamos los números.

De esta manera podemos ver que los números decimales entre 0 y 31 se pueden representar con números binarios de cinco posiciones. Cada una de estas posiciones recibe el nombre de BIT (abreviatura del ingles Binary Digit)

Con cinco bits escribimos:

0 = 00000
1 = 00001
.
.
.
7 = 00111
8 = 01000
.
.
.
16 = 10000
.
.
.
31 = 11111

Veamos un ejemplo: Representar en la base 2 el número 19



Expresamos esta igualdad del siguiente modo:

$$(10011)_2 = 1x2^4 + 0x2^3 + 0x2^2 + 1x2^1 + 1x2^0 = (19)_{10}$$

Ejercicio : ¿Cuántas lámparas serán necesarias para representar los números 0 y 255 en un sistema binario?

ANEXO: Charles Babbage - Un Pionero de la Computación Automática

Desde los tiempos de Leibnitz y Newton los matemáticos y, con más generalidad, los que en esos tiempos se conocían como filósofos naturales, se abocaron a la construcción de tablas o bien por medio de cálculos matemáticos (de multiplicación, de logaritmos, de senos y cosenos, etc.) o bien para registrar los resultados de sus mediciones (lluvias caídas en una determinada localidad, posición de la luna en cada medianoche del año, etc.). Por medio de estas tablas los científicos se comunicaron desde hace mucho tiempo, y lo hacen así para que sus experiencias puedan ser útiles a los demás.

Cuando se confecciona a mano una tabla que se deriva de la aplicación reiterada de una fórmula matemática, la experiencia demustra que se cometen muchos errores.

Charles Babbage (1791-1871) fue un matemático inglés que se interesó por la posibilidad de reemplazar a calculistas "falibles" por una máquina "infalible" que realizara los mismos cálculos sin equivocarse.

En 1823 presentó un trabajo en la Sociedad Real de Astronomía de Inglaterra, sobre "Observaciones respecto de la aplicación de máquinas para el cálculo de tablas de matemática", que le valió una medalla de oro, y que fue la base de su primer gran invento, la llamada Máquina de Diferencias.

Para ver cómo funcionaba esta máquina, debemos comprender primero qué es una tabla de diferencias. Consideremos, como ejemplo, la función:

$$y = 2x^2 + 3x + 4$$

La tabla de diferencias de esta función se construye de la siguiente manera:

x	y	D_1	D_2	D_3
0	4			
		5		
1	9		4	
		9		0
2	18		4	
		13		0
3	31		4	
		17		0
4	48		4	
		21		0
5	69		4	
		25		
6	94			

La columna D_1 (o primera diferencia) consiste de la diferencia entre valores sucesivos de la función y (5 = 9 - 4 = y(1) - y(0); 9 = 18 - 9 = y(2) - y(1); etc.). La columna D_2 (o segunda diferencia) se calcula como diferencia entre valores sucesivos de la columna D_1. Como en nuestro ejemplo la columna D_2 es constante, resulta que la tercera diferencia (D_3) y cualquier diferencia de orden superior debe ser igual a cero. La función de nuestro ejemplo, pertenece a una clase de funciones matemáticas muy importantes: los polinomios (funciones de la forma $a_n x^n + ... + a_1 x + a_0$, donde n es el grado). Un polinomio se comporta siempre de manera tal que existe un número n suficientemente grande para el cual la columna D_n es constante (prueben ustedes con $y = 2x^5 - x^3 + 1$ y con $y = x^5 + 1$).

En general: para un polinomio de grado n, la columna D_n es constante (a ustedes les deberá resultar D_5 constante en los dos casos). ¿Y para qué sirve una tabla así?. Para calcular sucesivos valores de un polinomio usando sólo sumas. Sigamos con el ejemplo de $y = 2x^2+3x+4$. Como $D_2 = 4$ constantemente, sabemos que el próximo valor de D_1 deberá ser $D_1 = 25 + 4 = 29$, pero entonces el próximo valor de y, o sea y(7) deberá satisfacer:

$$y(7) = y(6) + 29 = 94 + 29 = 123$$

Si reemplazan en el polinomio verán que efectivamente y(7) = 123.

Fue justamente éste el procedimiento que decidió mecanizar Babbage con su Máquina de Diferencias. Tal máquina debe consistir de un conjunto de registros que van tomando los sucesivos valores de cada columna.

Se les asigna a los registros los valores iniciales correctos y se construyen los sucesivos valores del polinomio, paso a paso, por medio de simples sumas.

En nuestro ejemplo necesitaremos sólo tres registros a los que llamaremos (1), (2) y (3). Inicialmente se le asigna a (1) el valor 4, a (2) el valor 5 y a (3) el valor 4. El ciclo para calcular el próximo valor de la función estará dado por: primero sume el registro (3) al registro (2); y sume el registro (2) al registro (1).

Babbage construyó una pequeña máquina con tres registros, que permitía tabular polinomios de grado 2 (como en nuestro ejemplo), y obtuvo apoyo oficial para desarrollar otra máquina que permitiría tabular polinomios de hasta grado 6. Esta máquina nunca fue completada, y 20 años después de obtenido se le retiró el apoyo oficial, y el proyecto fue abandonado.

La idea de Babbage fue recogida por un sueco, Pehr Georg Schentz (1785 - 1853). Schentz era editor de un diario, y fue su profesión la que le proveyó de una idea interesante: su máquina, muy similar a la que inventó Babbage, imprimía los valores que calculaba. Una copia de la máquina de Schentz fue usada hasta 1924 por el Registro General de las Personas, en Inglaterra, para calcular las tablas de esperanza de vida usadas por las compañías de seguro inglesas.

La Máquina de Diferencias tenía una función muy especializada, y no puede definirse de ninguna manera como un instrumento de uso general. Podía cumplir una única tarea (tabular un polinomio de hasta un grado dado). En ese sentido era un pasc atrás respecto de la máquina de Leibnitz, que sólo realizaba las cuatro operaciones matemáticas elementales, pero que podía servir para resolver una amplia gama de problemas matemáticos. Esta máquina es un intento, muy brillante por cierto, de automatizar un proceso específico, no es, en cambio, una máquina general capaz de automatizar el cálculo.

Hacia 1833, mientras se encontraba detenido en el desarrollo de la Máquina de Diferencias, Babbage concibió su obra maestra: la Máquina Analítica. Su concepción fue la de una computadora de propósito general, en el sentido moderno del término. Esta máquina fue el gran objetivo de su vida y trabajó en ella hasta su muerte, en 1871. La idea básica de esta máquina está tomada del dispositivo que J. M. Jacquard agregó a los telares en 1805, y que revolucionó la industria textil. El dispositivo de Jacquard automatizó el proceso del tejido de telas: para tejer una tela en un telar, el tejedor debe tener un plan o "programa", que le dice cuales hilos pasan por arriba de la trama y cuales pasan por debajo, para obtener el diseño buscado; también se le debe indicar cuándo debe repetir (o "iterar") el diseño patrón. El invento de Jacquard consistió en una serie de tarjetas con agujeros que describían el patrón deseado.

Los agujeros permitían que pasasen ganchos en algunos lugares. Estos ganchos hacían bajar a ciertos hilos (y a otros no), de modo tal que algunos quedaban por debajo de la trama, y otros por arriba, cuando pasaba la lanzadera.

Si se usan las mismas tarjetas, se puede variar el color de los hilos, pero el diseño patrón seguirá siendo el mismo. La analogía entre el invento de Jacquard y la Máquina Analítica es casi perfecta.

La Máquina Analítica consiste de dos partes:

1º) El depósito de números, para almacenar datos del problema como los valores generados por la máquina por medio de cálculos y

2º) El molino, un dispositivo que realiza operaciones aritméticas.

Como método de almacenamiento, Babbage propuso el uso de columnas de rueditas. Cada ruedita podía tomar un valor entre 0 y 9 y, por lo tanto podía representar un dígito decimal.

Una fórmula se describía por medio de ciertas operaciones sobre un conjunto dado de letras. Babbage propuso el uso separado de dos lotes de tarjetas de Jacquard: las tarjetas de operaciones que controlaban la acción del molino y especificaban qué operación ejecutar (suma, resta, etc.) y las tarjetas de variables, que controlaban la transferencia de número de y hacia el depósito de números.

Para cada paso se indica: sobre qué datos opera, qué operación se realiza, y adónde se debe almacenar el resultado numérico de la operación.

La Máquina Analítica es por lo tanto muy general. Sea cual fuere la fórmula que se desea calcular, se la debe especificar por medio de dos conjuntos de tarjetas. Una vez que se han ubicado estos dos conjuntos de tarjetas, la máquina está lista para procesar esa fórmula en particular.

Con un conjunto de tarjetas de operaciones dado, se puede recalcular la fórmula para distintos datos, cambiando las tarjetas de variables.

Todo esto suena muy moderno. El secreto de la Máquina Analítica consiste en la introducción de estos lotes de tarjetas, que hacen que las operaciones se realicen en forma automática y en una secuencia dada.

Lady Lovelace (Ada Byron, condesa de Lovelace, amiga y colaboradora de Babbage e hija de Lord Byron) dijo: "Podemos decir que de la misma manera que el telar de Jacquard teje flores y hojas, así la Máquina Analítica teje "diseños algebraicos". En esto hay mucha más originalidad que la que podía tener la Máquina de Diferencia".

Babbage tampoco pudo construir su Máquina Analítica. Su incapacidad para construir y usar sus dos grandes proyectos, destruyeron su interés en cualquier otra cosa, y le amargaron la vida.

El problema con el esquema de Babbage es que era muy ambicioso, y la precisión técnica de la ingeniería de mediados del Siglo XIX era muy inadecuada respecto de lo que Babbage requería.

ANEXO: Turing: Una Computadora con Lápiz y Papel (Extraído de un artículo del Dr. Sadosky, aparecido en Ciencia Nueva)

COMO CONSTRUIR UNA COMPUTADORA CON LAPIZ Y PAPEL

Una computadora automática es esencialmente un mecanismo capaz de obedecer órdenes y efectuar cálculos (aritméticos o lógicos). A pesar de las ideas corrientes que pueden inducir a vincular la noción de automatismo con la rapidez obtenida mediante dispositivos electrónicos, la velocidad no hace a la esencia del problema. Es más, una "máquina" sumamente sencilla que no es rápida pero tampoco es costosa, proporciona todos los elementos para comprender con la máxima claridad en qué consiste el funcionamiento automático, qué es un lenguaje interno y qué un lenguaje externo, qué es un programa y, en su conjunto, cómo se elabora la información que se le proporciona a la máquina y cómo esta devuelve la información procesada. Todos estos conceptos conservan su validez cuando se trata de las computadoras más sofisticadas y veloces que existen en el mercado.

LA MAQUINA DE TURING

Nos referimos a la llamada "máquina de Turing", creada con el propósito de discutir difíciles problemas de lógica matemática en el campo de la computabilidad y la teoría de la decisión, por el célebre lógico inglés Allan M. Turing (1912 - 1954) quien concibió este dispositivo en 1936, mucho antes de que se pensara en la construcción efectiva de computadoras automáticas electrónicas.

Todo el material que necesita disponer el lector para construir para sí mismo una máquina de Turing, es papel, preferentemente cuadriculado, un lápiz y una goma de borrar. Le aconsejamos que utilice ese material al mismo tiempo que lee este artículo para sacar de él efectivo provecho. Como dijimos antes, Turing ideó su "máquina" para encarar arduos problemas lógicos, pero nosotros nos limitaremos a usar el mismo dispositivo con propósitos didácticos, para resolver sencillísimos problemas aritméticos. La máquina de Turing consiste simplemente en una cinta infinita (en la práctica se considera una cinta indefinidamente prolongable en ambos sentidos) en la cual estan marcadas celdillas cuadradas de lado igual al ancho de la cinta; sobre la cinta se desplaza un cursor cuya abertura o visor equivale a una celdilla (figura 1). Para poder realizar con esta máquina -como con cualquier otra máquina- operaciones aritméticas, es necesario elegir un conjunto de símbolos y definir ciertas convenciones que permitan escribir los números con los símbolos elegidos y plantear las operaciones. Esos símbolos y esas convenciones constituyen lo que se llama el "lenguaje de la máquina".

En la Máquina de Turing utilizaremos los símbolos □ , | , * (y a veces también, como sustitutos del símbolo | , los símbolos: α , β , etc.) conviniendo que "escribir" un número en la máquina quiere decir poner en tantas celdillas como unidades tiene el número el símbolo | , colocando el símbolo * en la celdilla anterior al primer | de la izquierda y en la posterior al último | de la derecha.

Se conviene, además, que cuando se debe operar con dos números éstos se escriben uno a continuación del otro dejando una celdilla vacía (símbolo □) entre ambos.

Estos símbolos y estas convenciones que permiten la "escritura"
en la máquina constituyen la parte del lenguaje de la máquina llamada <u>ex</u>-
<u>terna</u>. El lenguaje se completa con el llamado lenguaje <u>interno</u> constituí-
do por el conjunto de símbolos que permiten "dar órdenes" a la máquina.
En la máquina de Turing el lenguaje interno está formado por
los símbolos A (inicial de "avanzar", que ordena que el cursor sea corrido
una celdilla de izquierda a derecha), R (inicial de "retroceder", que or-
dena mover el cursor de derecha a izquierda una celdilla), B (inicial de
"borrar", que indica que si aparece un símbolo en el visor debe ser borra-

do y que, generalmente, se omite poniendo simplemente el símbolo □ con lo cual se indica que se debe reemplazar con un blanco al símbolo que aparezca en el visor), ! (signo de admiración que se emplea para indicar que la operación ha terminado) y ? (signo que indica en este lenguaje interno que se ha cometido un error en el cálculo).

Una vez en posesión del "lenguaje" para poder operar es necesario conocer las reglas operativas, es decir disponer del algoritmo correspondiente a cada operación. (Un algoritmo es el conjunto de reglas mediante las cuales puede realizarse una operación aritmética o algebraica. Conocemos, por ejemplo, el algoritmo de la suma, el algoritmo de la raíz cuadrada, el algoritmo del máximo común divisor, etc.)

Si en la máquina hemos adoptado un sistema de escritura de los números diferente al sistema posicional de base 10, es lógico que el algoritmo de la suma tendrá una forma también diferente. Para que la máquina pueda operar automáticamente es necesario que le demos el algoritmo correspondiente: eso se hace escribiendo un programa que contendrá todas las órdenes que la máquina debe obedecer para poder sumar dos números cualesquiera.

COMÓ SE PRESENTA EL PROGRAMA

Para la máquina de Turing el programa se da en forma de una matriz (cuadro en cuyas columnas verticales figuran las órdenes que la máquina debe obedecer de acuerdo al símbolo que se encuentre en el visor y en las líneas horizontales las órdenes ordenadas según los sucesivos pasos del proceso) en la cual figuran las órdenes que la máquina debe obedecer para efectuar la operación. En cada lugar de la matriz pueden figurar tres símbolos: el primero corresponde a un símbolo del lenguaje externo con el cual debe reemplazarse al que se encuentra en el visor, el segundo es un símbolo del lenguaje interno que indica el movimiento que debe realizarse con el cursor y el tercero es el número escrito en sistema decimal que indica la línea horizontal a la cual se debe pasar para recibir las órdenes en el paso siguiente. No es necesario que en todos los casilleros figuren tres símbolos, pueden figurar dos o uno en ~~los~~ casos en que no deba cambiarse el signo que aparece en el visor y/o no resulte necesario cambiar de línea para buscar la orden para el paso siguiente. Puede haber casilleros de la matriz en los cuales no aparezcan órdenes, esos casilleros se llenan con un signo ? y si, en algún paso, se encuentra uno de esos símbolos, ello indica que se ha cometido algún error. Se coloca el símbolo ! en el casillero del programa correspondiente a la finalización de la operación.

Se conviene en comenzar la operación con el visor colocado en el primer | de la izquierda del segundo sumando.

Vamos a efectuar la operación 3 + 4. Escribimos estos sumandos de acuerdo a las convenciones establecidas, tal como aparece en la figura. En ella está señalada la posición inicial del cursor que convendremos corresponderá siempre al primer | del segundo sumando.

Para efectuar la operación de suma deben seguirse estrictamente las órdenes contenidas en la siguiente matriz programa:

	□	*	|
1	R2	□ R	R
2	A	R1	A3
3	A4	A	A
4	A	□ R7	□ R5
5	R	R6	?

	□	*	I
6	I R2	?	R
7	R	R8	?
8	*R9	?	R
9	!	?	?

Naturalmente lo que da a esta matriz el carácter de un verdadero algoritmo es que ella sirve para sumar dos números cualesquiera (es importante hacer notar que este programa no es el único posible y que pueden idearse otras matrices disponiendo la secuencia de órdenes de otra manera o partiendo de una posición inicial distinta)

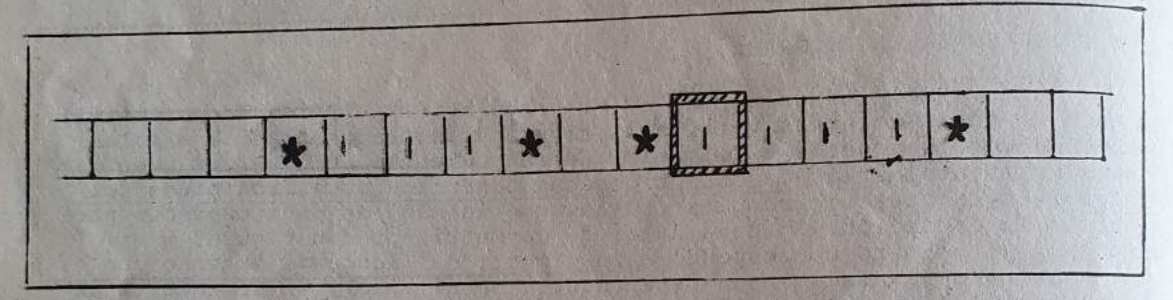


Fig. 1

UN EJEMPLO DE SUMA

Como ejemplo vamos a efectuar efectivamente y paso a paso la suma 3 + 4, a partir de la posición inicial indicada en la figura y siguiendo las instrucciones de la matriz dada.

Estamos en el primer paso (es decir en la horizontal señalada con el número 1) y frente a un I ; por consiguiente, de acuerdo con la órden R de la matriz, retrocedemos. Al retroceder encontramos un * en el visor y como el programa prescribe □ R, borramos el asterisco y retrocedemos. Vemos un blanco en el visor y como en el programa dice R2, retrocedemos y pasamos a la línea 2. Frente al visor hay ahora un * y como la orden dice R1, retrocedemos y volvemos a la línea 1. En el visor hay ahora un I ; la orden es R, luego retrocedemos y encontramos otro I y seguimos retrocediendo mientras aparezcan I en el visor. Cuando aparece un *, como la orden es □ R, borramos y retrocedemos. Encontramos un blanco, retrocedemos y pasamos a la línea 2. En el visor aparece entonces un blanco y la orden en la línea 2 es A; luego avanzamos. Volvemos a encontrar otro blanco y nuevamente avanzamos. Encontramos un I , avanzamos y pasamos a 3. Estamos frente a un I ; seguimos avanzando lo mismo que frente a los I o * que seguirán apareciendo. Al encontrar el primer blanco avanzamos y pasamos a la línea 4. Estamos frente a un blanco y, por tanto, avanzamos. Al encontrar un I tenemos la orden □ R5, luego borramos el palito I, retrocedemos y pasamos a 5. Estamos frente a un blanco, luego retrocedemos lo mismo que frente a los otros blancos que encontraremos sucesivamente. Al encontrar un * retrocedemos y pasamos a 6. Estamos frente a un I, retrocedemos y seguimos retrocediendo cada vez que encontramos I hasta que en-

contremos un blanco. La orden es entonces | R2, luego escribimos un | ,
retrocedemos y volvemos a la línea 2. Estamos frente a un blanco, en la
línea 2 la orden es A, luego avanzamos. Estamos frentea un | entonces
avanzamos y pasamos a 3. Como encontramos un | avanzamos y seguimos a
vanzando mientras encontremos | o *. Al encontrar un blanco avanzamos y
pasamos a la línea 4. Como encontramos nuevamente un blanco seguimos a-
vanzando hasta encontrar un | , entonces ponemos un blanco (tachamos el
|), retrocedemos y pasamos a 5. Estamos frente a un blanco y por tanto
retrocedemos y seguimos retrocediendo hasta encontrar un *. Frente al
* retrocedemos y pasamos a 6. Frente al | retrocedemos y seguimos retro-
cediendo frente a los | que aparecen en el visor hasta encontrar un blan-
co, entonces ponemos un | , retrocedemos y pasamos a 2. Estamos frente
a un blanco y de acuerdo con la orden de la línea 2 avanzamos y encon-
tramos un |. Frente al | avanzamos y pasamos a 3. Estamos frente a un
| y por tanto avanzamos así como frente a todos los | que van aparecien-
do y frente a *. Al encontrar un blanco volvemos a avanzar y pasamos a
4. Estamos frente a un blanco y volvemos a avanzar asi como frente a los
sucesivos blancos que aparecen. Al encontrar un | lo borramos (es decir
ponemos un blanco), retrocedemos y pasamos a 5. Estamos frente a un blan-
co, luego retrocedemos así como frente a los sucesivos blancos que van
apareciendo. Al encontrar un * retrocedemos y pasamos a 6. Estamos fren-
te a un |, retrocedemos y seguimos retrocediendo frente a los | que apa-
recen en el visor, hasta encontrar un blanco. Frente al blanco la orden
es | R2, luego ponemos un | , retrocedemos y pasamos a 2. Estamos frente
a un blanco, luego avanzamos. Encontramos un | , avanzamos y pasamos a
3. Estamos frente a un | , avanzamos asi como frente a los sucesivos |
que irán apareciendo en el visor. Cuando aparece un * también avanzamos
y pasamos a 4. Frente al blanco que aparecerá en el visor avanzamos y
seguimos avanzando frente a los sucesivos blancos hasta que aparezca un
| . Borramos ese | , retrocedemos y pasamos a 5. Estamos frente a un
blanco, de acuerdo con la orden R, retrocedemos y seguimos retrocedien-
do frente a los sucesivos blancos que aparecerán en el visor. Al encon-
trar un * retrocedemos y pasamos a 6. Estamos frente a un |, retrocede-
mos así como frente a los sucesivos | que irán apareciendo. Al encontrar
un blanco ponemos un | , retrocedemos y pasamos a 2. Estamos frente a un
blanco, luego avanzamos y estamos frente a un |, entonces avanzamos† así
como frente a los sucesivos | que se irán presentando en el visor y tam-
bién avanzamos al encontrar un *. Al encontrar un blanco, avanzamos y
pasamos a 4. Estamos frente a un blanco, avanzamos así como frente a los
sucesivos blancos. Ahora llegamos a encontrar un * que reemplazamos por
un blanco (borrando), luego retrocedemos y pasamos a 7. Estamos frente a
un blanco y retrocedemos así como frente a los sucesivos blancos que apa-
recerán en el visor. Al encontrar un *, retrocedemos y pasamos a 8. Es-
tamos frente a un | , luego retrocedemos así como frente a los sucesivos
| que aparecerán en el visor. Al encontrar un blanco ponemos un *, re-
trocedemos y pasamos a 9. Estamos frente a un blanco y como en la casi-
lla correspondiente de la matriz hay un signo ! eso significa que la o-
peración está terminada. En efecto en la cinta sólo han quedado escritos
siete signos, precedidos y seguidos por *, es decir, el número siete de
acuerdo a la convención adoptada.

UN EJEMPLO DE PRODUCTO

Para multiplicar, suponiendo como ejemplo que la operación a rea-
lizar fuera 3 x 4, la posición inicial sería la misma de la figura con
que ilustramos la suma.

En la matriz programa de la multiplicación que damos a continua-
ción figura una columna más que las que aparecían en las matrices de la
suma, correspondiente a un nuevo símbolo $\propto$ .Como se verá, al realizar e-
fectivamente una multiplicación, este aparece en sustitución del signo
en los pasos intermedios del cálculo.

La matriz programa para la multiplicación es la siguiente:

† (y pasamos a 3. Estamos frente a un |, avanzamos)

	□	*	|	∝
1	R	□ R3	□ R2	?
2	?	□ 1	R	?
3	A7	A	∝ R4	| A
4	?	R5	R	?
5	| R6	□	R	R3
6	*	A	A	R3
7	A	□ 9	□ R8	?
8	R	?	∝ R4	?
9	R	!	□ R	?

Es aconsejable efectuar operaciones de acuerdo con los programas dados para comprender mejor el mecanismo operacional. Para hacerlo se puede trabajar con la "cinta" dibujada sobre un papel e indicando a cada paso la posición de un cursor imaginario mediante una flecha, dibujada con lápiz que señale en cada paso dónde se encuentra el visor y se vaya borrando a medida que el cursor supuestamente se traslada, o bien haciendo una cinta cuadriculada de cartulina sobre la cual pueda efectivamente desplazarse un visor del mismo material.

También será un ejercicio provechoso confeccionar otros programas similares a los dados. Por ejemplo, preparar una matriz para la suma a partir de la posición inicial del visor en el último del segundo sumando o una matriz para la resta que permita restar dos números escribiendo primero el minuendo y luego el sustraendo, etc.

Este es un ejercicio lógico que permitirá comprender muy bien el tipo de trabajo que debe realizar un programador cuando prepara el programa de cálculos para una computadora electrónica de cualquier tipo.

OBSERVACIONES SOBRE EL PROGRAMA

El lector que, aceptando nuestra sugerencia, ha realizado la suma y el producto de los números tres y cuatro habrá notado que ha actuado como un autómata, pues en cada paso se le ha indicado en qué fila debía actuar, qué desplazamiento (a la derecha o a la izquierda) debía efectuar, qué símbolo había que escribir o borrar, sin exigirle ningún razonamiento. Es fácil concebir que un dispositivo mecánico, electromecánico o electrónico, puede hacer exactamente lo que ha hecho el operador obediente. Conviene señalar que una vez especificado el programa, el operador actúa automáticamente, pero también que un programador es una persona que en conocimiento del procedimiento que se debe seguir para hacer un cálculo prescribe las instrucciones que debe obedecer el mecanismo. El lector puede intentar hacer programas. Por ejemplo multiplicar de otra manera haciendo que en lugar de sumar 4 veces "paquetes" de 3 unidades como se ha hecho en el ejemplo desarrollado haya que sumar 3 veces "paquetes" de 4 unidades.

LA MAQUINA UNIVERSAL DE TURING

El lector habrá podido construir uno o varios programas y hacerlos funcionar con su máquina de papel. Ahora bien, cada programa sirvió para resolver una clase de problemas determinada. Para construir un programa hubo que fabricar una matriz que luego debía ser "ejecutada" paso a paso sobre la cinta de papel. Por más complicado que sea el problema, la matriz-programa será grande pero en definitiva, finita. Los números y símbolos de la matriz pueden codificarse como "tiras" de ceros y unos sin

inconvenientes (serán sucesiones finitas siempre) y "grabarse" sobre la cinta, la misma que nos servía de "memoria" de los datos a manipular.

De este modo la máquina universal de Turing tiene la cinta dividida en dos secciones principales: a la izquierda está la descripción codificada de la matriz programa a ejecutar y a la derecha la sucesión de símbolos que dicho programa va a encontrar al inspeccionar la cinta. La máquina universal se construye entonces de modo que el visor vaya y venga entre las dos secciones izquierda y derecha de la cinta. Mediante un complicado sistema de marcas y señales, la máquina universal lleva la cuenta de la línea de la matriz (codificada en el lado izquierdo) que se está "ejecutando". Turing demostró que el efecto de la máquina universal sobre la sucesión de datos del lado derecho es el mismo que se produciría con la "matriz-programa" externo que se usó en la parte anterior operando sobre esos mismos datos.

En otras palabras, si codificamos la matriz de la suma en el lado izquierdo y cargamos los números a sumar en el lado derecho, obtendremos en el lado derecho el resultado correcto. Este es el principio de la máquina de programa almacenado.

4.

ANEXO: La Máquina de Von Neumann

John Von Neumann (1903 - 1957) fue un matemático y físico húngaro que vivió radicado en E.E.U.U.

Sus intereses matemáticos eran muy amplios, y abarcaban tanto a la lógico como a la hidrodinámica (que es la rama de la física matemática que estudia el movimiento de los fluidos, como el agua, por ejemplo)

Entre las décadas de 1930 y 1940, Von Neumann se había puesto a estudiar problemas de hidrodinámica que eran imposibles de resolver con las herramientas de cálculo con que se contaba en ese momento.

Un encuentro fortuito con gente del grupo que en ese momento se encontraba desarrollando la ENIAC en Moore School, Filadelfia, hizo que Von Neumann hiciera converger su interés por la lógica con su interés por la física matemática.

En sucesivas reuniones con el grupo de Moore School, Von Neumann planteó los requisitos que debía reunir una máquina capaz de resolver los problemas de hidrodinámica que le interesaban. Se decidió que no era factible modificar la ENIAC para que cumpliera con esos requisitos, los cuales serían objeto de un desarrollo posterior.

El 30 de junio de 1945 Von Neumann, como consultor del grupo de Moore School, presentó un informe que resumía las ideas del grupo, y que es uno de los trabajos fundacionales sobre los que se basa la computación moderna. Este trabajo es el primer borrador de la EDVAC.

En ese trabajo se explicita que una computadora ejecuta escencialmente funciones lógicas, y que los aspectos electrónicos son sólo subsidiarios. Además se formula un estudio preciso y detallado de las funciones e interacciones mutuas entre las diferentes partes de una computadora.

Dice Von Neumann:

1.1) Las consideraciones que siguen se refieren a la estructura de un sistema de computación automático, digital de muy alta velocidad.

1.2) Un sistema de computación automático es un dispositivo que puede ejecutar instrucciones que realicen cálculos de un considerable orden de complejidad. Las instrucciones que gobiernan esta operación se deben dar al dispositivo de manera absolutamente exhaustiva. Estas incluyen toda la información numérica requerida para resolver el problema. Estas instrucciones se le deben dar de alguna manera comprensible para el dispositivo: perforadas en tarjetas o cintas de teletipo, impresas magnéticamente en cintas de acero, etc. Todos estos procedimientos requieren el uso de algún código que permita expresar la definición lógica y algebraica del problema que se está considerando. Una vez que el dispositivo recibe estas instrucciones, debe estar en condiciones de resolverlo completamente, sin ninguna intervención humana posterior. Después de ejecutar estas operaciones, el dispositivo debe registrar los resultados en alguno de los medios antes descriptos. Los resultados serán datos numéricos. Más adelante enumero los órganos de la máquina:

2.2) PRIMERO: Deberá realizar frecuentemente las cuatro operaciones elementales. Es razonable que contenga entonces órganos especializados para realizar estas operaciones. Sin embargo todas las operaciones deben ser supervisadas por una parte específica del dispositivo: la central aritmética (CA).

2.3) El control lógico del dispositivo, es decir el control de la correcta sucesión de sus operaciones, lo debe realizar un órgano central de control (CC). Este órgano debe controlar la ejecución en orden de las instrucciones, sinimportarle cuáles son esas instrucciones en particular (debe ser un dispositivo elástico, independiente del problema). Las instrucciones se deben almacenar en forma independiente de este órgano.

2.4) TERCERO: a) Cualquier dispositivo que realice una gran cantidad de cálculos debe tener una gran MEMORIA....
b) Las instrucciones que gobiernan un problema complicado pueden ser muchas. Este material debe ser recordado....
La tercera parte específica la constituye la memoria total (M).

2.5) Las tres partes específicas CA, CC (a las que en conjunto llamaremos C) y M corresponden a las neuronas asociativas del sistema nervioso. Corresponde discutir ahora los equivalentes de las neuronas sensitivas y de las motoras. Estas serán los órganos de entrada y salida del dispositivo. El dispositivo debe poder mantenerse en contacto con algún medio externo de registración como los vistos en el punto (1.2), al que llamaremos R.....

2.6) CUARTO: El dispositivo debe poseer órganos que transfieran la información de R a C y a M. Estos órganos forman la unidad de entrada: I. Se verá que es preferible que todas las transferencias se realicen de R a M, nunca a C.

2.7) QUINTO: El dispositivo debe poseer órganos para transferir datos de C y M a R. Estos órganos forman su unidad de salida: Ø. Se verá que es preferible, otra vez, que las transferencias se realicen de M a R, nunca de C a R.

En el mismo trabajo Von Neumann propuso un repertorio de instrucciones para esa computadora, y en una nota posterior trabajó en la programación detallada de una rutina para ordenar un conjunto de números y de otra para intercalar dos conjuntos ordenados de números. Esto representa un mojón, ya que es la primera ilustración absolutamente desarrollada del concepto de programa almacenado.

Dejó además instaurado el concepto de modo de operación en serie (es decir, se inspecciona y ejecuta una instrucción por vez) en oposición con la ENIAC, que operaba en paralelo (es decir: se ejecutaban simultáneamente muchas cosas).

Pese a que la tradición indicaba que se debían construir máquinas digitales con números representados en sistema decimal, se propuso para la EDVAC que los números se representaran en sistema binario. Las razones expuestas fueron: la extrema simplicidad y rapidez con las cuales se pueden realizar las operaciones elementales, el hecho de que los circuitos electrónicos son binarios por naturaleza, y finalmente el hecho de que las porciones de control de una computadora no son aritméticas sino lógicas, y la lógica es un sistema binario (basado en dos valores: verdadero o falso).

En marzo de 1946, se comenzó a desarrollar en el Instituto de Estudios Avanzados de Princeton una máquina que se conoció como "máquina de Von Neumann". Los órganos de esta máquina corresponden a los descriptos en el primer borrador de la EDVAC.

El conjunto de instrucciones de esta máquina permitía: ejecutar todas las operaciones aritméticas elementales, transferir datos de la memoria a la unidad aritmética y viceversa, y cambiar la dirección de la memoria de la próxima instrucción en forma condicional (cambiarla sólo si se cumple cierta condición sobre el valor de un número), o incondicional (cambiarla en cualquier circunstancia).

Para comprender este concepto veamos lo siguiente: podemos pensar que las posiciones de la memoria de la máquina se pueden ennumerar en orden creciente. El órgano de control se puede construir de manera tal que normalmente, después de ejecutar la instrucción almacenada en la dirección n-ésima, procede a inspeccionar y ejecutar la instrucción almacenada en la dirección (n+1)-ésima; salvo que de alguna manera se cambie la dirección de la próxima instrucción, y ya no sea n+1. Finalmente, existía un conjunto de instrucciones referidas a los órganos de entrada y salida.